Interactive and Computational Methods for Molecular Modelling Applied to the Bacterial Ribosome

vorgelegt von
Diplom-Informatiker
Ingolf E. Sommer

Vom Fachbereich 13 – Informatik
der technischen Univerität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
– Dr. rer. nat. –
genehmigte Dissertation

Promotionsausschuß:

Vorsitzender: Prof. Dr.-Ing. Günter Hommel
Berichter: Dr. Richard Brimacombe
Berichter: Prof. Dr. H.U. Lemke

Tag der wissenschaftlichen Aussprache: 6.1.2000

Berlin 2000

D 83

Interactive and Computational Methods for Molecular Modelling Applied to the Bacterial Ribosome

Dissertation
Ingolf E. Sommer
August 1999

Max-Planck-Institut für molekulare Genetik
AG Ribosomen
Technische Universität Berlin
Fachbereich Informatik

Supervision: Prof. Dr. H.U. Lemke
Dr. R. Brimacombe
Extramural Advice: Prof. Dr. S.C. Harvey

Acknowledgments

This work originated at the *Max-Planck-Institut für molekulare Genetik* and would have been impossible without Dr. Richard Brimacombe, whom I thank explicitely for many stimulating discussions and for the positive collaboration.

I thank Prof. Dr. Steve Harvey for providing the possibility for several fruitful visits to his Lab, for the exchange of ideas there and in Berlin and continued by email.

I also thank Dr. Pavel Baranov, Prof. Dr. Jessica Hodgins, Petra Philips, Dr. Robert Tan and Margaret VanLoock for inspiring discussions. I am grateful to Dr. Florian Mueller at our institute for passing on a plenitude of experience in modelling and programming. I appreciate the excellent technical support of Fritz Rammer and Frank Mante from our computing centre.

I am grateful to Prof. Dr. Günter Hommel and Prof. Dr. Heinz Lemke for their interest in this interdisciplinary work, for their advice and for accepting this thesis at the *Fachbereich Informatik* of the *Technische Universität Berlin*.

Berlin, August 1999

Abstract

A computational multi-scale modelling approach for the refinement of large biomolecules was designed and then applied to the structure of the bacterial ribosome.

Algorithms were developed, allowing defined groups of atoms to be clustered into rigid objects, which greatly reduces the number of parameters in the molecular dynamics approach and thus speeds up the computational process considerably (clustered molecular dynamics).

The energy potential function, which is used in molecular dynamics to describe structural details of a particle, was extended to include terms that describe high-level biochemical constraints resulting from cross-linking techniques and cryo-electron microscopy.

High- and low-level features of the potential function were specified, and the clustered molecular dynamics technique was integrated into the interactive model building process, to establish a physico-chemically plausible structure of the bacterial ribosome.

Kurzfassung

Rechnerische Methoden zur effektiven Modellierung von großen Biomolekülen wurden entwickelt und dann auf die Struktur des bakteriellen Ribosoms angewendet.

Basierend auf der im Englischen molekulardynamisch (molecular dynamics) genannten Technik wurden Algorithmen entworfen, um definierte Gruppen von Atomen zu starren Clustern zusammenzufassen; dadurch läßt sich die Anzahl der Parameter stark vermindern und damit der Rechenprozeß beschleunigen (Clustered Molecular Dynamics (CMD)).

Die im molekulardynamischen Ansatz zur Beschreibung der strukturellen Details von Partikeln verwendete Energiepotentialfunktion wurde um zwei Terme erweitert. Diese berücksichtigen gröbere biochemische Strukturzusammenhänge, die sich aus Cross-linking Techniken und aus der Cryo-Elektronen-Mikroskopie ergeben.

Grob- und detailstrukturelle Eigenschaften der Potentialfunktion wurden spezifiziert und die CMD Technik wurde in den interaktiven Modellbildungsprozeß einbezogen um ein Strukturmodell des bakteriellen Ribosoms zu generieren. Dieses weist einen hohen Plausibilitätsgrad auf, da es den zugrundegelegten biochemisch-physikalischen Anforderungen entspricht.

Contents

Introduction

In living organisms translation of genetic information into protein takes place on the ribosome which is therefore a key-particle in molecular genetics. The ribosome is a particle consisting of more than 200000 atoms which are organized in strands of ribonucleic acid and several proteins folded in a complex manner. Understanding the function of such a particle requires knowledge about its structure.

In order to investigate the structure of ribosomes different physical and biochemical techniques have been applied to gather information:

Electron microscopy, for instance, delivers density images of the complete particles with a limited resolution (section 4.3.4). X-ray crystallography and nuclear magnetic resonance techniques yield detailed information about substructures with up to atomic resolution (sections 4.3.2 and 4.3.3).

Biochemical techniques such as *sequencing*, *cross-linking*, and *foot-printing* are used to analyze the strands of ribonucleic acid, to find intra-molecular distances and spatial relations of neighbouring segments, as well as interaction mechanisms of ribonucleic acid and protein (sections 4.2.1 to 4.2.3).

Some of the information is available for several species which allows the comparison of the differences in sub-structures; some of the information has been obtained for particular species only. Accordingly, a great amount of data of different origin, quality, and resolution exists.

The aim of several scientific groups is to find a model of the ribosome in agreement with these data, e.g. [BASS88, SWN88, MH94]. The degree of information provided by the data available up to now is so different for various regions of the ribosome that some of them can be modelled up to atomic resolution, whereas others can only be outlined as a sketch of the structure.

Computers are powerful tools for building such models, because large quantities of structural data can be stored, visualized, and processed. Processing can either be controlled interactively, by the biochemist supported by the computer, or computationally based on theoretical models only.

The aim of this thesis is to develop computational methods for refining models of particles towards biochemical and physicochemical correctness in a general manner and to apply them to the ribosome as a specific particle.

Accordingly, the first part of the thesis concentrates on the elaboration of computational methods, based on the principles of classical physics and mechanics (chapter 1). The methods are designed for the simulation of large molecular systems under the aspect of energy minimization, providing the capability of incorporating data with different levels of resolution, and to assure a close link to the interactive modelling process. For the latter purpose, a stepwise iterative approach is sug-

gested, starting with interactive manipulation at a coarse scale, refining structural details by computation, and giving feedback for optimizing the interactive approach (chapter 2).

These computational processes are based on known structural properties of bioparticles. Since the particles consist of strands which may be interconnected, constituents of the strands as well as interconnected parts of the strands can be considered as building blocks. The building blocks are groups of atoms in certain conformations that repeatedly occur. According to the size of these groups, several levels can be distinguished.

The reason why a lot more information is needed for understanding the structural properties of particles is explained for the ribosome and its specific constituents, ribonucleic acid and protein, in the second part of the thesis.

The main aim of this second part is to improve the plausibility of generated models of the ribosome by developing and applying defined criteria for the computational refinement of structural details, whereby the computational refinement has to meet the requirements determined by the resolution of data available for the various regions of the particle and by conserving the structural details down to the atomic scale that were already incorporated into the interactively generated model.

Accordingly, this part contains a brief review of structure and function of the ribosome (chapter 3), a description of the various methods for gaining information and data (chapter 4), and a definition of the necessary input data for the various kinds of computations to be carried out (chapter 5). Based on this information the computational approach developed is integrated into the model building process (chapter 6).

Part I

Computational Molecular Modelling

Chapter 1

Computational Molecular Biophysics

Methods to computationally prognose and analyze the structure and behaviour of chemical particles range from quantum mechanics to large scale abstractions [MH87, LB92, LB93, Rap95].

The focus of chapter 1 is on the computational treatment of large scale molecules and particles of biological relevance. For this purpose some basic assumptions are made (section 1.1), and a complete framework is introduced to define (section 1.2), simulate (sections 1.3 to 1.5) and improve (section 1.6) the structures of these particles.

Whereas in this chapter the computational, non-interactive generation of structure is described, the following chapter shows methods to manipulate structure interactively and how to combine interactive manipulation with computational techniques.

These considerations form the basis for the application to the structure of the ribosome in the second part.

1.1 Model Assumptions

1.1.1 Approaches

Modelling at the atomic level is based on a comprehensive description of the particles involved. To be precise, such a description must be based on quantum mechanics. The molecular dynamics approach offers a framework, operating with classical Newtonian forces on atoms instead. It replaces the quantum picture of interactions arising from overlapping electron clouds by a system of point masses, coupled by exotic "springs". Therefore the generated models should not be taken literally on the small scale, but offer an efficient means to compute large scale structures.

The inter-atomic relations are described by a set of forces, or correspondingly an energy potential. Classical Newtonian mechanics allows the equations of motion of the atoms to be derived, and, by integration over time, their trajectories to be computed. It is thus possible to find stable structures by tracking the physical motion within particles.

A related but slightly different approach is to directly operate minimization algorithms on the same energy potential function, using the atom-positions as parameters. The difference is conceptual. Depending on the minimization methods used, potentially un-physical movement of atoms will occur in the minimization process. For a review of minimization methods in this context see [Sch92].

1.1.2 Computational Atoms

Atoms are the basic computational units within the molecular dynamics framework. The molecules and particles treated computationally consist of atoms. These are approximated as charged point-masses in three-dimensional space, with a certain position, charge, mass, type, and – for visualization purposes – also extent. The atom type, such as hydrogen, helium, lithium, etc., yields information about the atoms' mass and extent. These remain constant, whereas the atoms' charge and position may change with respect to time.

Extended Atoms

For large systems it is often useful to combine several atoms into one *extended atom* [BBO+83]. Most often hydrogen atoms are combined with neighbouring heavier atoms. Sometimes even larger groups of atoms are modelled as one *pseudo-atom* [MTH94], or groups of atoms can be clustered into rigid objects.

1.1.3 Energy Potential

Intra-atomic phenomena are not considered. Instead, atoms are related to each other by forces, such as bonding forces, electrostatic forces, and Van-der-Waals interaction forces. These relations are used to describe intra- and inter-atomic effects in a simplified manner. Formally the forces are derived from an energy potential, defined in detail in section 1.2.

1.1.4 Limitations

Atomic systems obey quantum laws rather than classical Newtonian laws. An estimate of approximation error is thus in place:

A simple test of the validity is the de Broglie thermal wavelength [Erc97, HM86] defined as

$$\Lambda = \sqrt{\frac{2\pi h^2}{M k_b T}}, \tag{1.1}$$

where M is the atomic mass, T is the temperature, k_b is the Boltzmann constant, and h is Planck's Wirkungsquantum. The classical approximation is justified if $\Lambda \ll a$, where a is the mean-nearest-neighbour separation. For instance, for liquids at the triple point, Λ/a is about 0.1 for light elements such as Lithium and Argon, decreasing further for heavier elements. The classical approximation is poor for very light systems such as hydrogen, helium or neon. The systems considered as

applications clearly do not belong to this latter category. Quantum effects also gain importance when the temperature is very low. This is also not the case for the biological systems considered as applications here.

1.2 Energy Potential Function

Energy potential functions numerically describe observed relations between atoms. Thus they allow different structures to be evaluated and given structures to be improved.

First, atom-to-atom relations are considered, which are relevant for local features of structure. Nevertheless, these local structural properties imply consequences on the intermediate and overall spatial arrangement of a particle. Later, the atom-to-atom relations are extended to include coarser information, such as cross-linking data (cf. section 4.2.5) and electron microscopy densities (cf. section 4.3.4). A complete potential is finally generated by summation of the potential functions generated by the different relations.

1.2.1 Covalent Bond Energy

Atom a_i is assumed to be a point mass at position $\boldsymbol{a}_i = (a_{i,x}, a_{i,y}, a_{i,z})^T$. Two atoms a_i and a_j can be connected by a bond b_{ij}. The bond length is the distance between the two atoms

$$r_{ij} = |\boldsymbol{r}_{ij}| = |\boldsymbol{a}_j - \boldsymbol{a}_i| = \sqrt{(a_{j,x} - a_{i,x})^2 + (a_{j,y} - a_{i,y})^2 + (a_{j,z} - a_{i,z})^2}, \tag{1.2}$$

where $\boldsymbol{r}_{ij}$ is the distance vector from $\boldsymbol{a}_i$ to $\boldsymbol{a}_j$.

Covalent bonds are modelled as springs, keeping the atoms in the correct distance $r_{0,ij}$. Thus, a force constant $k_{b_{ij}}$ is defined analogous to a spring constant for each bond. It depends on the type of the atoms involved.

Let B be the set of all defined bonds within the observed particle. As a convention, B contains b_{ij}, if it also contains b_{ji}, with the same force constant $k_{b_{ji}} = k_{b_{ij}}$, and desired length $r_{0,ji} = r_{0,ij}$. The energy within a single bond is $E_{b_{ij}}$, and the total bond energy is defined as:

$$E_B = \sum_{\substack{b_{ij} \in B \\ i<j}} E_{b_{ij}} = \sum_{\substack{b_{ij} \in B \\ i<j}} k_{b_{ij}} (r_{ij} - r_{0,ij})^2. \tag{1.3}$$

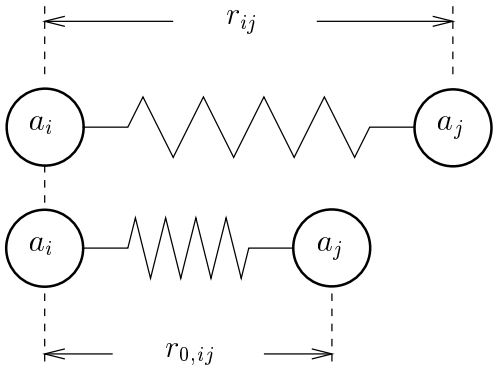


Figure 1.1: Bonds are modelled as springs, striving to keep atoms a_i and a_j in the desired distance $r_{0,ij}$.

1.2.2 Bond Angle Energy

The angle θ_{ijk} included by two bonds b_{ij} and b_{ik} at atom a_i is usually important. It is defined by

$$\cos\theta_{ijk} = \frac{\boldsymbol{r}_{ij}^T\,\boldsymbol{r}_{ik}}{r_{ij}\,r_{ik}} = \frac{(\boldsymbol{a}_j - \boldsymbol{a}_i)^T(\boldsymbol{a}_k - \boldsymbol{a}_i)}{r_{ij}\,r_{ik}}. \tag{1.4}$$

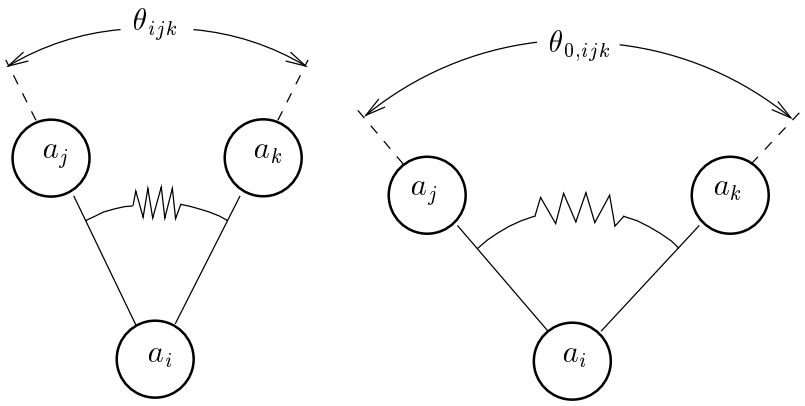


Figure 1.2: Bond angles θ_{ijk} are defined as the angle between bonds b_{ij} and b_{ik}. Energetically they are modelled as springs, pushing the three involved atoms a_i, a_j, and a_k so that the desired angle $\theta_{0,ijk}$ is taken.

To measure how good an angle corresponds to its desired value $\theta_{0,ijk}$, another spring-like construction is used, with a force constant $k_{\theta_{ijk}}$. Note that the angle relation is symmetric in the two latter indices only.

Again, Θ denotes the set of all relevant angles θ_{ijk} within the modelled system. As a convention, when $\theta_{ijk} \in \Theta$, then $\theta_{ikj} \in \Theta$ and when $k_{\theta_{ikj}} = k_{\theta_{ijk}}$, then $\theta_{0,ikj} = \theta_{0,ijk}$. With $E_{\theta_{ijk}}$ being the energy within a single bond angle, the complete bond angle potential is:

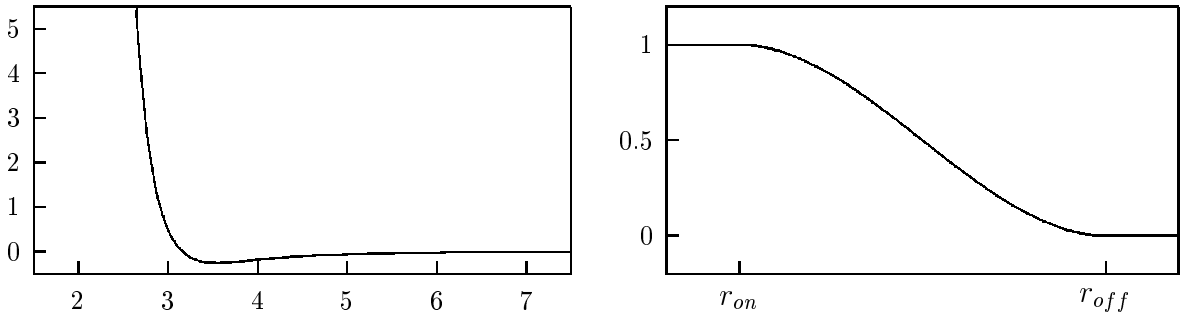
$$E_\Theta = \sum_{\substack{\theta_{ijk}\in\Theta \\ j<k}} E_{\theta_{ijk}} = \sum_{\substack{\theta_{ijk}\in\Theta \\ j<k}} k_{\theta_{ijk}}(\theta_{ijk} - \theta_{0,ijk})^2. \tag{1.5}$$

1.2.3 Van-der-Waals Energy

The Lennard-Jones potential is used to model Van-der-Waals interactions of non-bonded atoms. The potential energy decreases rapidly with increasing distance of the atoms involved (figure 1.3 (a)). For practical purposes a switching function sw is introduced, neglecting Van-der-Waals effects at a distance greater than r_{off}.

Let $vdw_{i,j}$ be the Van-der-Waals interaction of two atoms a_i and a_j, $E_{vdw_{i,j}}$ the corresponding Lennard-Jones energy, and VDW the set of all interactions taken into account. Like bond and angle-energies, the Van-der-Waals relation is symmetrical, and as a convention $vdw_{ji} \in VDW$, when $vdw_{ij} \in VDW$, with $A_{ji} = A_{ij}, B_{ji} = B_{ij}$:

$$E_{VDW} = \sum_{\substack{vdw_{i,j}\in VDW \\ i<j}} E_{vdw_{i,j}} = \sum_{\substack{vdw_{i,j}\in VDW \\ i<j}} \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}}\right) sw(r_{ij}^2, {r_{\text{on}}}^2, {r_{\text{off}}}^2). \tag{1.6}$$



(a) (b)

Figure 1.3: (a) Typical Lennard-Jones function describing the repulsive potential of atoms. Energy in kcal/mole is plotted on the y-axis, distance in Ångstrom on the x-axis. The parameters are $A = 10^6$, and $B = 10^3$. (b) The switching function $sw(r, r_{\text{on}}, r_{\text{off}})$ smoothly interpolates from 1 to 0.

A_{ij} and B_{ij} are force constants specifying the repulsional and attractional potential, and sw is defined as

$$sw(r, r_{\text{on}}, r_{\text{off}}) = \begin{cases} 1 & \text{for} \quad r \leq r_{\text{on}} \\ \frac{(r_{\text{off}} - r)^2 (r_{\text{off}} + 2r - 3r_{\text{on}})}{(r_{\text{off}} - r_{\text{on}})^3} & \text{for} \quad r_{\text{on}} < r \leq r_{\text{off}} \\ 0 & \text{for} \quad r > r_{\text{off}} \end{cases} . \tag{1.7}$$

This switching function is chosen such (figure 1.3 (b)), that it smoothly interpolates from 1 to 0, and differentiates as zero at the points r_{on} and r_{off}.

1.2.4 Distance Measurements – Cross-Linking

There are biochemical techniques that allow distance measurements. These can yield precise information such as: Atom a_i must be no farther than a certain distance from atom a_j. They can be much more imprecise: One atom within set of atoms X must be within a certain distance d of an atom within another set Y. Depending on the distances and the number of atoms in the sets, these measurements can be used to build or to verify 3D structures. For the ribosome, data of this type are generated with the cross-linking method (cf. section 4.2.5). Therefore this type of potential is referred to as *abstract cross-link*. Let CL be the set of such cross-link constraints $cl_l = (X_l, Y_l, d_l)$, an abstract cross-link potential then is suggested as:

$$E_{CL} = \sum_{cl_l \in CL} E_{cl_l} = \sum_{cl_l \in CL} k_{cl_l} \left(\chi\Big(\min_{a_i \in X_l, a_j \in Y_l} (r_{ij}) \; - d_l \Big) \right)^2 , \tag{1.8}$$

where

$$\chi(x) = \begin{cases} 0 & \text{for} \quad x \leq 0 \\ -x^3 + 2x^2 & \text{for} \quad 0 < x \leq 1 \\ x & \text{for} \quad 1 < x \end{cases} . \tag{1.9}$$

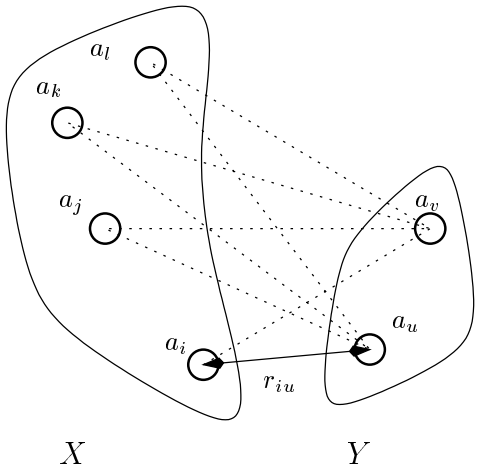


Figure 1.4: For the cross-linking energy potential, distances are computed from atoms within set X (here $= \{a_i, a_j, a_k, a_l\}$) to atoms within set Y (here $= \{a_u, a_v\}$). The minimal distance (here r_{iu}) is then compared to the desired cross-link length d.

The potential E_{cl_l} is comparing the minimal distance of atoms within sets X_l and Y_l with the desired distance d_l (figure 1.4). E_{cl_l} is continuous. It is differentiable, except for positions where several atoms of either set X_l and Y_l are at minimal distance. The same holds for E_{CL} as the summation. The problem of partial undifferentiability can be overcome by improving the potential definition theoretically, such that abrupt changes at the according positions do not occur. Here it was defined empirically and the consequences are also treated empirically, in order to be able to work with this simplification.

1.2.5 Density Constraints

For some states of certain molecules densities, i.e. the masses within volume elements at given positions in three-dimensional space, are known. Depending on the size of the molecule, the generation techniques, and on the sophistication of their application, the specification and resolution of density data can vary greatly.

In this context, there are two fundamentally different approaches to evaluate atoms in density computationally (figure 1.5): One approach computes the total mass of the atoms per volume element and compares it to the given density of the volume. The other approach evaluates for each atom how well it fits into density, without considering other atoms. The first approach is physically more precise. The second implies forces acting on the atoms and is thus easier to handle within the molecular dynamics framework introduced in the following section. It is also more robust in handling models that do not incorporate all relevant atoms (yet). Since this was very much the case during the modelling of the ribosome, only the latter approach will be pursued here.

The density data are assumed to be specified for uniform volume elements, called *voxels*. The given resolution defines the size of a unit voxel. The space that the molecule is computed on is then divided into such voxels, and for each voxel a density value is specified: low value meaning few atoms, high value many atoms.

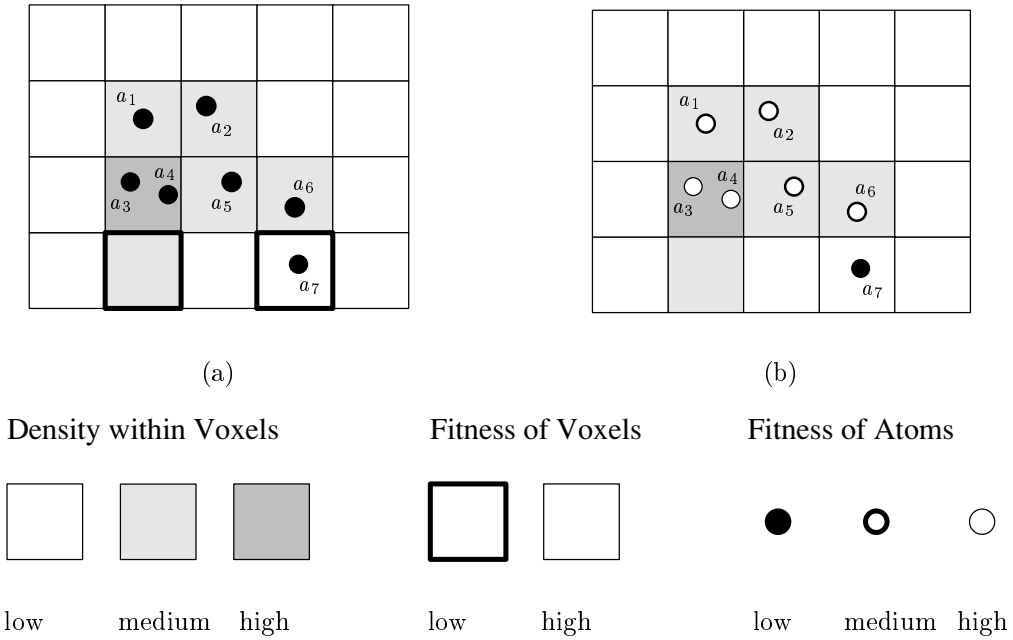


Figure 1.5: Two approaches for evaluating an atomic model according to density information: (a) The atomic mass is computed per voxel and compared to its density in order to compute a voxel fitness function. (b) Each atom is evaluated how well it fits into density to compute an atomic fitness function.
Note that approach (a) implies a malus for high-density voxels with few atoms as well as for low-density voxels with many atoms. Approach (b) penalizes atoms in low-density regions only, but not high density regions without atoms.

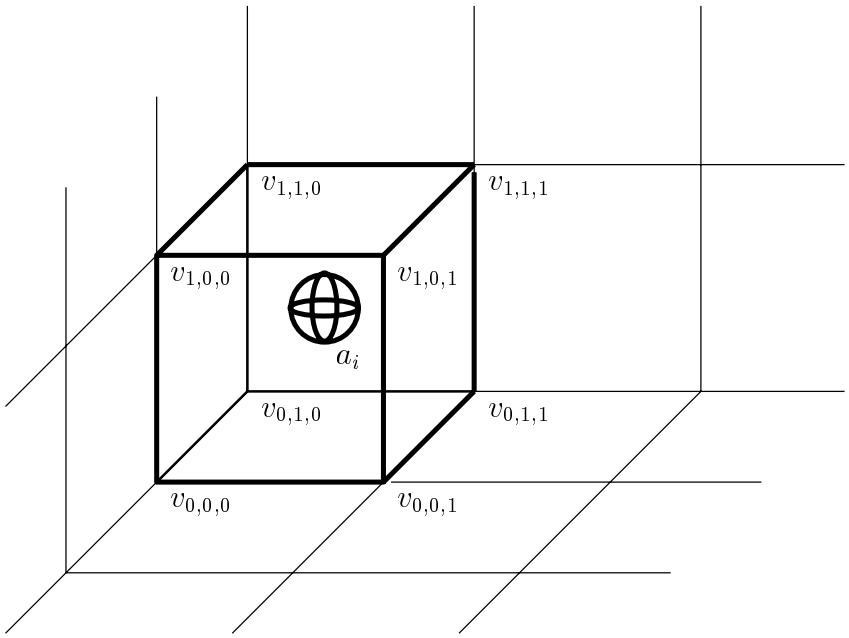


Figure 1.6: A regular grid is used to describe molecular shape, such as the information gained by 3D electron microscopy. Each grid point is associated with a density value v. A fitting function $\text{fit}(a_i)$ is then interpolated for each atom a_i from the surrounding grid values.

A regular grid with grid length l in each direction is used to index density values. One grid point is assumed at the origin; the points adjacent in x-direction are $\ldots, V_{-2l,0,0}, V_{-l,0,0}, V_{0,0,0}, V_{l,0,0}, V_{2l,0,0}, \ldots$, in y-direction $\ldots, V_{0,-l,0}, V_{0,0,0}, V_{0,l,0}, \ldots$, and in z-direction $\ldots, V_{0,0,-l}, V_{0,0,0}, V_{0,0,l}, \ldots$. Each grid-point lies in the centre of a voxel and has the density value $v_{il,jl,kl}$ associated, where i, j, k are integers.

How well an atom a_i fits into density, is interpolated between values of neighbouring grid points. Specify $s_{i,x} = \lfloor \frac{a_{i,x}}{l} \rfloor$ and $g_{i,x} = s_{i,x} + l$ as the x-indices of the grid-point neighbouring in x-direction and analogously $s_{i,y}$, $g_{i,y}$, $s_{i,z}$, and $g_{i,z}$. The fit of a single atom is then interpolated as:

$$\begin{aligned}
\mathrm{fit}(a_i) = \frac{1}{l^3} \quad \Big(\quad & (a_{i,x} - s_{i,x})(a_{i,y} - s_{i,y})(a_{i,z} - s_{i,z}) \quad v_{s_{i,x},s_{i,y},s_{i,z}} \\
+ \quad & (a_{i,x} - s_{i,x})(a_{i,y} - s_{i,y})(g_{i,z} - a_{i,z}) \quad v_{s_{i,x},s_{i,y},g_{i,z}} \\
+ \quad & (a_{i,x} - s_{i,x})(g_{i,y} - a_{i,y})(a_{i,z} - s_{i,z}) \quad v_{s_{i,x},g_{i,y},s_{i,z}} \\
+ \quad & (a_{i,x} - s_{i,x})(g_{i,y} - a_{i,y})(g_{i,z} - a_{i,z}) \quad v_{s_{i,x},g_{i,y},g_{i,z}} \\
+ \quad & (g_{i,x} - a_{i,x})(a_{i,y} - s_{i,y})(a_{i,z} - s_{i,z}) \quad v_{g_{i,x},s_{i,y},s_{i,z}} \\
+ \quad & (g_{i,x} - a_{i,x})(a_{i,y} - s_{i,y})(g_{i,z} - a_{i,z}) \quad v_{g_{i,x},s_{i,y},g_{i,z}} \\
+ \quad & (g_{i,x} - a_{i,x})(g_{i,y} - a_{i,y})(a_{i,z} - s_{i,z}) \quad v_{g_{i,x},g_{i,y},s_{i,z}} \\
+ \quad & (g_{i,x} - a_{i,x})(g_{i,y} - a_{i,y})(g_{i,z} - a_{i,z}) \quad v_{g_{i,x},g_{i,y},g_{i,z}} \quad \Big). \qquad (1.10)
\end{aligned}$$

Let S be the set of all atoms considered for density, the energy potential is then:

$$E_S = \sum_{a_i \in S} (\mathrm{fit}_{\max} - \mathrm{fit}(a_i))^2 \qquad (1.11)$$

$\mathrm{fit}_{\max} = \max_{a_i \in S} \mathrm{fit}(a_i)$ is constant, equal to the maximum fit of a single atom. The potential is continuous. It is differentiable, except at atom positions falling on the grid lines.

The pretreatment of the voxel based density data influences the potential. For the ribosome this will be discussed in section 5.5.

1.2.6 Total Energy

With the individual components defined in the previous sections, the total potential energy E is summed as:

$$E = E_B + E_\Theta + E_{VDW} + E_{CL} + E_S. \qquad (1.12)$$

A stable conformation of a molecule is now a local minimum of the potential function. The deeper the minimum, the more stable the conformation. A plausible structure is a minimum with a low energetic value – not necessary a global minimum.

1.2.7 Comparison With Standard Potentials

The potential function defined in the previous section was used in the application to the ribosome. It is an extended version of a simplified CHARMM [BBO+83] potential function which is one of the established standards. The CHARMM potential

has additional angle energy, electrostatic, hydrogen-bonding and constraint terms, but lacks the shape and cross-link extensions.

Obviously, the potential per se defines the structure of the particle in question (cf. chapter 5). For the computational process, each term of the potential function needs to be calculated. Apart from that the potential has little influence on the computational methods.

1.3 Forces

The force acting on an atom is defined as the negative derivative of the potential energy with respect to the atom position. These atomic forces are required for the subsequent simulation and minimization processes. For reference purposes they are listed here.

1.3.1 Covalent Bond Forces

The force implied by bonds on a single atom a_i can be derived as the negative of the gradient of the bond potential E_B, with respect to the three variables $a_{i,x}, a_{i,y}$, and $a_{i,z}$. In the following description, all formulae will be given for the x-coordinate $a_{i,x}$ only – the y and z coordinates follow analogously.

The force induced by bond lengths acting on atom a_i is

$$F_{B,i} = -(\frac{\partial E_B}{\partial a_{i,x}}, \frac{\partial E_B}{\partial a_{i,y}}, \frac{\partial E_B}{\partial a_{i,z}})^T. \tag{1.13}$$

The derivative $\partial E_B / \partial a_{i,x}$ can be computed as

$$\frac{\partial E_B}{\partial a_{i,x}} = \sum_{\text{all } b_{ij} \text{ joining atom } a_i} \frac{\partial E_{b_{ij}}}{\partial a_{i,x}}. \tag{1.14}$$

$E_{b_{ij}}$ is differentiated according to the chain-rule $\frac{\partial E_{b_{ij}}}{\partial a_{i,x}} = \frac{\partial E_{b_{ij}}}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial a_{i,x}}$ which results in:

$$\frac{\partial E_{b_{ij}}}{\partial a_{i,x}} = 2k_{b_{ij}} \frac{(r_{ij} - r_{0,ij})(a_{i,x} - a_{j,x})}{r_{ij}}. \tag{1.15}$$

1.3.2 Bond Angle Forces

Similarly, the force operating on atom a_i generated by angle constraints is defined as:

$$F_{\Theta,i} = -(\frac{\partial E_\Theta}{\partial a_{i,x}}, \frac{\partial E_\Theta}{\partial a_{i,y}}, \frac{\partial E_\Theta}{\partial a_{i,z}})^T, \tag{1.16}$$

where the potential E_Θ derived by the atoms coordinate is:

$$\frac{\partial E_\Theta}{\partial a_{i,x}} = \sum_{\theta_{ijk} \in \Theta} \frac{\partial E_{\theta_{ijk}}}{\partial a_{i,x}} + \sum_{\theta_{jik} \in \Theta} \frac{\partial E_{\theta_{jik}}}{\partial a_{i,x}}. \tag{1.17}$$

The first term is induced by angle constraints that centre on a_i, the second term by constrains with a_i on one of their flanks. By the chain-rule the derivatives are:

$$\frac{\partial E_{\theta_{ijk}}}{\partial a_{i,x}} = -2k_{\theta_{ijk}}(\theta_{ijk}-\theta_{0,ijk})\left(\frac{r_{ij}r_{ik}(2a_{i,x}-a_{j,x}-a_{k,x})}{r_{ij}^2 r_{ik}^2\sqrt{1-\left(\frac{\boldsymbol{r}_{ij}^T\boldsymbol{r}_{ik}}{r_{ij}r_{ik}}\right)^2}} - \frac{\frac{\boldsymbol{r}_{ij}^T\boldsymbol{r}_{ik}}{r_{ij}r_{ik}}(r_{ik}^2(a_{i,x}-a_{j,x})+r_{ij}^2(a_{i,x}-a_{k,x}))}{r_{ij}^2 r_{ik}^2\sqrt{1-\left(\frac{\boldsymbol{r}_{ij}^T\boldsymbol{r}_{ik}}{r_{ij}r_{ik}}\right)^2}}\right) \quad (1.18)$$

and

$$\frac{\partial E_{\theta_{jik}}}{\partial a_{i,x}} = -2k_{\theta_{jik}}(\theta_{jik}-\theta_{0,jik})\frac{r_{ji}r_{jk}(a_{k,x}-a_{j,x})-\frac{\boldsymbol{r}_{ji}^T\boldsymbol{r}_{jk}}{r_{ji}r_{jk}}r_{jk}^2(a_{i,x}-a_{j,x})}{r_{ji}^2 r_{jk}^2\sqrt{1-\left(\frac{\boldsymbol{r}_{ji}^T\boldsymbol{r}_{jk}}{r_{ji}r_{jk}}\right)^2}}. \quad (1.19)$$

The optical complexity of the expressions is only due to the differentiation of the arcus-cosine function that defines the angle θ_{ijk}.

1.3.3 Van-der-Waals Forces

The Van-der-Waals force imposed on atom a_i is analogously defined as

$$F_{VDW,i} = -\left(\frac{\partial E_{VDW}}{\partial a_{i,x}}, \frac{\partial E_{VDW}}{\partial a_{i,y}}, \frac{\partial E_{VDW}}{\partial a_{i,z}}\right)^T, \quad (1.20)$$

where the derivative of the potential E_{VDW}, as defined in (1.6), is linearly decomposed into the sum of derivatives of the pair-wise energies $E_{vdw_{i,j}}$

$$\frac{\partial E_{VDW}}{\partial a_{i,x}} = \sum_{vdw_{i,j}\in VDW}\frac{\partial E_{vdw_{i,j}}}{\partial a_{i,x}} \quad (1.21)$$

which are

$$\frac{\partial E_{vdw_{i,j}}}{\partial a_{i,x}} = \left[\left(\frac{6B_{ij}}{r_{ij}^7}-\frac{12A_{ij}}{r_{ij}^{13}}\right)sw(r_{ij}^2,r_{on}^2,r_{off}^2) + \left(\frac{A_{ij}}{r_{ij}^{12}}-\frac{B_{ij}}{r_{ij}^6}\right)sw'(r_{ij}^2,r_{on}^2,r_{off}^2)2r_{ij}\right]\frac{a_{i,x}-a_{j,x}}{r_{ij}}. \quad (1.22)$$

The derivative of the switching function sw is

$$sw'(r,r_{on},r_{off}) = \begin{cases} 0 & \text{for} \quad r \le r_{\text{on}} \\ \frac{6(r_{off}-r)(r_{on}-r)}{(r_{off}-r_{on})^3} & \text{for} \quad r_{\text{on}} < r \le r_{\text{off}} \\ 0 & \text{for} \quad r > r_{\text{off}} \end{cases}. \quad (1.23)$$

Therefore no force is generated by a single Van-der-Waals interaction pair with atoms at a distance greater than the cut-off radius r_{off}.

1.3.4 Cross-Link Forces

The force generated by cross-links, that is acting on atom a_i, is defined as

$$F_{CL,i} = -\left(\frac{\partial E_{CL}}{\partial a_{i,x}}, \frac{\partial E_{CL}}{\partial a_{i,y}}, \frac{\partial E_{CL}}{\partial a_{i,z}}\right)^T , \tag{1.24}$$

which, again, is decomposed into the forces generated by single cross-links

$$\frac{\partial E_{CL}}{\partial a_{i,x}} = \sum_{cl_l \in CL} \frac{\partial E_{cl_l}}{\partial a_{i,x}} \tag{1.25}$$

which are computed as

$$\frac{\partial E_{cl_l}}{\partial a_{i,x}} = \begin{cases} \chi'(r_{ij} - d_l)\frac{a_{i,x} - a_{j,x}}{r_{ij}} & \text{for} \quad a_i \in X_l \ \wedge \ \min\limits_{a_j \in Y_l}(r_{ij}) = \min\limits_{a_k \in X_l, a_j \in Y_l}(r_{kj}) \\ \chi'(r_{ij} - d_l)\frac{a_{i,x} - a_{j,x}}{r_{ij}} & \text{for} \quad a_i \in Y_l \ \wedge \ \min\limits_{a_j \in X_l}(r_{ij}) = \min\limits_{a_j \in X_l, a_k \in Y_l}(r_{kj}) \\ 0 & \text{else} \end{cases} . \tag{1.26}$$

The derivative of the cut-off function χ, defined in (1.9) is

$$\chi'(x) = \begin{cases} 0 & \text{for} \quad x \leq 0 \\ -3x^2 + 4x & \text{for} \quad 0 < x \leq 1 \\ 1 & \text{for} \quad 1 < x \end{cases} . \tag{1.27}$$

The undifferentiability of the energy potential function at certain positions poses a theoretical problem here. In practice the function is forced to be differentiable by additional assumptions.

1.3.5 Density Constraint Forces

The force pushing atom a_i into a given density element is

$$F_{S,i} = -\left(\frac{\partial E_S}{\partial a_{i,x}}, \frac{\partial E_S}{\partial a_{i,y}}, \frac{\partial E_S}{\partial a_{i,z}}\right)^T , \tag{1.28}$$

with

$$\frac{\partial E_S}{\partial a_{i,x}} = 2(\text{fit}(a_i) - \text{fit}_{\max}) \, \frac{\partial \, \text{fit}(a_i)}{\partial a_{i,x}} \tag{1.29}$$

and the derivative of the fitting function:

$$\begin{aligned} \frac{\partial \, \text{fit}(a_i)}{\partial a_{i,x}} = \frac{1}{l^3} \quad \Big(\quad & (a_{i,y} - s_{i,y})(a_{i,z} - s_{i,z}) \quad v_{s_{i,x}, s_{i,y}, s_{i,z}} \\ + \quad & (a_{i,y} - s_{i,y})(g_{i,z} - a_{i,z}) \quad v_{s_{i,x}, s_{i,y}, g_{i,z}} \\ + \quad & (g_{i,y} - a_{i,y})(a_{i,z} - s_{i,z}) \quad v_{s_{i,x}, g_{i,y}, s_{i,z}} \\ + \quad & (g_{i,y} - a_{i,y})(g_{i,z} - a_{i,z}) \quad v_{s_{i,x}, g_{i,y}, g_{i,z}} \\ - \quad & (a_{i,y} - s_{i,y})(a_{i,z} - s_{i,z}) \quad v_{g_{i,x}, s_{i,y}, s_{i,z}} \\ - \quad & (a_{i,y} - s_{i,y})(g_{i,z} - a_{i,z}) \quad v_{g_{i,x}, s_{i,y}, g_{i,z}} \\ - \quad & (g_{i,y} - a_{i,y})(a_{i,z} - s_{i,z}) \quad v_{g_{i,x}, g_{i,y}, s_{i,z}} \\ - \quad & (g_{i,y} - a_{i,y})(g_{i,z} - a_{i,z}) \quad v_{g_{i,x}, g_{i,y}, g_{i,z}} \quad \Big) \quad . \end{aligned}$$

1.4 Molecular Dynamics

The underlying potential function is an approximation to nature. Accordingly there is no point in following trajectories to a high degree of accuracy. Instead, the goal is to reproduce certain time-space dependent conformations to a sufficient degree of accuracy.

1.4.1 Newton's Laws Reformulated

Kinematics

The aim is to track a system of atoms over time. Let $S = \{a_1, ...a_n\}$ be this system of n atoms to be tracked.

Each atom a_i has a constant mass $a_{i,m}$ and a position

$$\boldsymbol{a} = (a_{i,x}, a_{i,y}, a_{i,z})^T \tag{1.30}$$

in a three-dimensional Cartesian vector space. Its velocity is defined as

$$\dot{\boldsymbol{a}} = \frac{d\,\boldsymbol{a}}{dt} = (\frac{da_{i,x}}{dt}, \frac{da_{i,y}}{dt}, \frac{da_{i,z}}{dt})^T, \tag{1.31}$$

its acceleration as

$$\ddot{\boldsymbol{a}} = \frac{d^2\,\boldsymbol{a}}{dt^2} = (\frac{d^2a_{i,x}}{d^2t}, \frac{d^2a_{i,y}}{d^2t}, \frac{d^2a_{i,z}}{d^2t})^T. \tag{1.32}$$

Join all the atomic coordinates into one vector to obtain

$$\boldsymbol{s} = (a_{1,x}, a_{1,y}, a_{1,z}, a_{2,x}, a_{2,y}, a_{2,z}, \ldots, a_{n,z})^T, \tag{1.33}$$

and analogously the atomic velocities and accelerations

$$\dot{\boldsymbol{s}} = (\dot{a}_{1,x}, \dot{a}_{1,y}, \dot{a}_{1,z}, \dot{a}_{2,x}, \dot{a}_{2,y}, \dot{a}_{2,z}, \ldots, \dot{a}_{n,z})^T, \tag{1.34}$$

$$\ddot{\boldsymbol{s}} = (\ddot{a}_{1,x}, \ddot{a}_{1,y}, \ddot{a}_{1,z}, \ddot{a}_{2,x}, \ddot{a}_{2,y}, \ddot{a}_{2,z}, \ldots, \ddot{a}_{n,z})^T. \tag{1.35}$$

Dynamics

Newton's laws relate force and motion – allowing the system to be tracked over time; within this context they state:

Law of Inertia An atom a_i remains in its state of rest or motion in a straight line at a constant velocity $\dot{\boldsymbol{a}}_i$ unless it is obliged by acting forces to change this state.

Law of Motion An atom a_i that is subjected to a force $\boldsymbol{F}_i$ is given an acceleration $\ddot{\boldsymbol{a}}_i$ so that $\boldsymbol{F}_i = a_{i,m}\ddot{\boldsymbol{a}}_i$, where $a_{i,m}$ is the constant inertial mass of the atom.

Law of Action and Reaction The forces exerted by two atoms onto each other are equal in magnitude and opposite in direction.

Defined by the potential in section 1.2, the force acting on each atom solely depends on the position of the atom itself and on the positions of the other atoms:

$$\boldsymbol{F}_i = f_i(\boldsymbol{s}). \tag{1.36}$$

The atoms' acceleration is related to the force by the law of motion $\boldsymbol{F}_i = a_{i,m}\ddot{\boldsymbol{a}}_i$, so that:

$$\ddot{\boldsymbol{a}}_i = F_i/a_{i,m} = f_i(\boldsymbol{s})/a_{i,m}. \tag{1.37}$$

Considering all atoms simultaneously and introducing a vector valued function g, this system of second order differential equations can be reformulated into

$$\ddot{\boldsymbol{s}} = g(\boldsymbol{s}) = \begin{pmatrix} f_1(\boldsymbol{s})/a_{1,m} \\ f_2(\boldsymbol{s})/a_{1,m} \\ f_3(\boldsymbol{s})/a_{1,m} \\ f_4(\boldsymbol{s})/a_{2,m} \\ f_5(\boldsymbol{s})/a_{2,m} \\ f_6(\boldsymbol{s})/a_{2,m} \\ \vdots \\ f_{3n}(\boldsymbol{s})/a_{n,m} \end{pmatrix} \tag{1.38}$$

and is generally referred to as the (Newtonian) equations of motion.

For the given forces, this system of equations is clearly not integrable directly (symbolically), thus numerical integration techniques need to be applied. Given starting positions and velocities for the atoms, the problem turns out to be a classical initial value problem.

1.4.2 Numerical Integration

Selection Criteria

The idea for any routine solving initial value problems numerically is to rewrite the infinitesimal $d\boldsymbol{s}$ and dt as finite steps, and to derive algebraic formulae for the change in the function, when increasing the dt by one stepsize h. In the limit of making the stepsize h very small, a good approximation to the underlying differential equation is achieved. There are several types of classical algorithm for numerically integrating differential equations, including Runge-Kutta methods, Richardson extrapolation, and predictor-corrector methods [Wal90, PTVF97]. Which one is best adapted to the problem?

The heaviest component of the integration process is the evaluation of the forces. Thus any integration method requiring more than one such calculation per timestep must, to be useful, deliver a proportionate increase in the size of the timestep while maintaining the same accuracy.

Because of the strongly repulsive force at short distances based on the Lennard-Jones term in the potential, there is an upper limit in the timestep. Therefore, the Runge-Kutta methods are not competitive in this context: Since they are not able to increase the timestep beyond this limit, they generally require several evaluations per timestep.

On the other hand the underlying potential is – at a few places only – not differentiable. Thus the generated forces may be discontinuous, excluding Richardson extrapolation which heavily relies on the function's smoothness.

Two kinds of algorithms are frequently used in molecular dynamics simulations: Leap-frog type integration and predictor-corrector methods.

Leapfrog Type Numerical Integration

The Leap-frog type and the Verlet numerical integration schemes are presented in this section. They are algebraically equivalent and, unlike the predictor-corrector methods presented subsequently, have an intuitive appeal.

Verlet The Verlet formula follows from the Taylor expansion of the coordinate vector $\boldsymbol{s}(t)$

$$\boldsymbol{s}(t+h) = \boldsymbol{s}(t) + h\dot{\boldsymbol{s}}(t) + \frac{h^2}{2}\ddot{\boldsymbol{s}}(t) + O(h^3). \tag{1.39}$$

Add the corresponding expansion for $\boldsymbol{s}(t-h)$

$$\boldsymbol{s}(t-h) = \boldsymbol{s}(t) - h\dot{\boldsymbol{s}}(t) + \frac{h^2}{2}\ddot{\boldsymbol{s}}(t) - O(h^3) \tag{1.40}$$

and rearrange to obtain:

$$\boldsymbol{s}(t+h) = 2\boldsymbol{s}(t) - \boldsymbol{s}(t-h) + h^2\ddot{\boldsymbol{s}}(t) + O(h^4). \tag{1.41}$$

The truncation error is of order $O(h^4)$ because the $O(h^3)$ terms cancel. The acceleration values $\ddot{\boldsymbol{s}}(t)$ are simply the evaluation of the force-function $g(\boldsymbol{s}(t))$ at time t.

Leapfrog Insert the right side of the Taylor expansion of

$$\dot{\boldsymbol{s}}(t+h/2) = \dot{\boldsymbol{s}}(t) + \frac{h}{2}\ddot{\boldsymbol{s}}(t) + O(h^2) \tag{1.42}$$

into the rearranged Taylor expansion of $\boldsymbol{s}(t+h)$

$$\boldsymbol{s}(t+h) = \boldsymbol{s}(t) + h\Big[\dot{\boldsymbol{s}}(t) + \frac{h}{2}\ddot{\boldsymbol{s}}(t)\Big] + O(h^3) \tag{1.43}$$

to get the leapfrog integration formulae:

$$\begin{aligned} \boldsymbol{s}(t+h) &= \boldsymbol{s}(t) + h\dot{\boldsymbol{s}}(t+h/2) + O(h^3) && (1.44)\\ \dot{\boldsymbol{s}}(t+h/2) &= \dot{\boldsymbol{s}}(t-h/2) + h\ddot{\boldsymbol{s}}(t) + O(h^3). && (1.45) \end{aligned}$$

The velocity term (1.45) is the subtraction of the expansion of $\dot{\boldsymbol{s}}(t-h/2)$ from $\dot{\boldsymbol{s}}(t+h/2)$ in (1.42).

The evaluation of coordinates and velocities at different timesteps does not cause a problem. The velocity at the current time can be estimated by

$$\dot{\boldsymbol{s}}(t) \approx \dot{\boldsymbol{s}}(t-\frac{h}{2}) + \frac{h}{2}\ddot{\boldsymbol{s}}(t). \tag{1.46}$$

Predictor-Corrector Numerical Integration

Like the leapfrog methods in the previous section, the predictor-corrector method aims at solving $\ddot{\boldsymbol{s}} = g(\boldsymbol{s})$ numerically.

At each timestep, the past position, velocity, and acceleration values are used to extrapolate a future position value. For this extrapolated position, the accelerations and velocities are evaluated and used to correct the position estimate. Both the extrapolation and the correction are done by polynomial interpolation. Thus a three step process is obtained:

Predictor (P) step: The past position values $\boldsymbol{s}(t), \boldsymbol{s}(t-h), \boldsymbol{s}(t-2h), \ldots$ and their derivatives $\dot{\boldsymbol{s}}(t), \ldots$ and $\ddot{\boldsymbol{s}}(t), \ldots$ are known. Use polynomial extrapolation to estimate a position $\boldsymbol{s}_{est}$ and velocity $\dot{\boldsymbol{s}}_{est}$ at time $t+h$:

$$P: \quad \boldsymbol{s}_{est}(t+h) = \boldsymbol{s}(t) + h\dot{\boldsymbol{s}}(t) + h^2 \sum_{i=1}^{k-1} \alpha_i \ddot{\boldsymbol{s}}(t+(1-i)h) \tag{1.47}$$

$$P': \quad \dot{\boldsymbol{s}}_{est}(t+h) = \frac{1}{h}\boldsymbol{s}_{est}(t+h) - \frac{1}{h}\boldsymbol{s}(t) + h \sum_{i=1}^{k-1} \alpha'_i \ddot{\boldsymbol{s}}(t+(1-i)h) \tag{1.48}$$

A specification of the α_i and α'_i is given by (1.53) below.

Evaluation (E) step: The forces at the estimated new position are computed yielding the acceleration

$$E(\boldsymbol{s}_{est}): \quad \ddot{\boldsymbol{s}}(t+h) = g(\boldsymbol{s}_{est}(t+h)). \tag{1.49}$$

Corrector (C) step: Finally the acceleration values are used to correct the position and velocity estimates

$$C: \quad \boldsymbol{s}(t+h) = \boldsymbol{s}(t) + h\dot{\boldsymbol{s}}(t) + h^2 \sum_{i=1}^{k-1} \beta_i \ddot{\boldsymbol{s}}(t+(2-i)h) \tag{1.50}$$

$$C': \quad \dot{\boldsymbol{s}}(t+h) = \frac{1}{h}\boldsymbol{s}(t+h) - \frac{1}{h}\boldsymbol{s}(t)) + h \sum_{i=1}^{k-1} \beta'_i \ddot{\boldsymbol{s}}(t+(2-i)h) \tag{1.51}$$

Optionally, a second evaluation can be performed to replace the acceleration values at the estimated position $\boldsymbol{s}_{est}(t+h)$ with their counterparts at the more precise new position $\boldsymbol{s}(t+h)$

$$E(\boldsymbol{s}): \quad \ddot{\boldsymbol{s}}(t+h) := g(\boldsymbol{s}(t+h)). \tag{1.52}$$

Polynomial extrapolation and correction are possible in different ways. Popular schemes are the Adams-Bashford method for extrapolation, and the Adams-Moulton for correction, suggesting the choice of the parameters α_i, α'_i, and β_i, β'_i respectively to satisfy

$$\sum_{i=1}^{k-1}(1-i)^q \alpha_i = \frac{1}{(q+1)(q+2)} \qquad \sum_{i=1}^{k-1}(1-i)^q \alpha'_i = \frac{1}{q+2} \tag{1.53}$$

$$\sum_{i=1}^{k-1}(2-i)^q \beta_i = \frac{1}{(q+1)(q+2)} \qquad \sum_{i=1}^{k-1}(2-i)^q \beta'_i = \frac{1}{q+2} \tag{1.54}$$

for $q = 0, \ldots, k-2$. These linear equations can be solved, e.g., for $k = 4$, the parameters are:

$$\begin{array}{llll} \alpha_1 = \frac{19}{24}, & \alpha_2 = -\frac{10}{24}, & \alpha_3 = \frac{3}{24}, & \\ \beta_1 = \frac{3}{24}, & \beta_2 = \frac{10}{24}, & \beta_3 = -\frac{1}{24}, & \end{array} \qquad \begin{array}{lll} \alpha'_1 = \frac{27}{24}, & \alpha'_2 = -\frac{22}{24}, & \alpha'_3 = \frac{7}{24}, \\ \beta'_1 = \frac{7}{24}, & \beta'_2 = \frac{6}{24}, & \beta'_3 = -\frac{1}{24}. \end{array}$$

Instead of using multiple past steps $\boldsymbol{s}(t), \boldsymbol{s}(t-h), \boldsymbol{s}(t-2h), \boldsymbol{s}(t-3h), \ldots$, multiple derivatives at the same time can be used $\boldsymbol{s}(t), \dot{\boldsymbol{s}}(t), \ddot{\boldsymbol{s}}(t), \dddot{\boldsymbol{s}}(t), \ldots$ to predict future values. The former approach is called multistep predictor-corrector method, and the latter multivalue predictor-corrector. They can be proven to be equivalent [PTVF97]. The multivalue approach has the advantage of simplifying adaptive timesteps. However, the multistep approach seems more natural to use, since higher order derivatives are not meaningful per se in the physical model.

Adaptive Timestep

Usually molecular dynamics simulations run with a fixed timestep – the integration of the differential equation is then non-adaptive. In this work, the precise tracking of particles was less important than the energy minimization aspect, rendering adaptive methods interesting (section 1.6.2) – especially in combination with the clustering described in section 1.5.

Adapting the timestep brings up two issues: I) how to control the timestep, and II) how to compute the past values used by the predictor-corrector algorithm, if the timestep is changed.

I) Controlling the timestep can basically be attacked in two ways: Either theoretical limits are computed from the potential – or, much easier in practice, empirical values are chosen and the process is supervised to remain stable.

II) Changing the timestep from h to h' makes the predictor-corrector algorithm lose its necessary data from the past $\ddot{\boldsymbol{s}}(t-h')$, $\ddot{\boldsymbol{s}}(t-2h'), \ldots$. This problem can be solved by polynomial interpolation.

1.4.3 Initial Values

For the molecular dynamics simulation processes, velocity values can be initialized by randomization, imitating heat movement of the particles. If no good starting model is given, then the initialization of the position vectors causes problems. Randomization of the positions does not necessarily result in stable structures. Often, models from previous simulation runs are selected and combined.

Alternatively geometric initialization approaches can be followed. In combination with interactive improvement, this was also the starting point for the ribosome model (cf. chapters 5 and 6).

1.5 Clustered Molecular Dynamics

In the preceeding sections general atomic properties of biomolecules were defined, and it was shown how to simulate systems of such atoms. In addition higher level structural properties can be exploited. The 3D structure of intra-molecular building blocks, consisting of several atoms, is often known in advance (cf. section 3.1.2). The number of parameters can be highly reduced by clustering the atoms within such a building block into a rigid object: instead of tracking many atomic positions, only the positions and orientations of considerably fewer objects need to be traced.

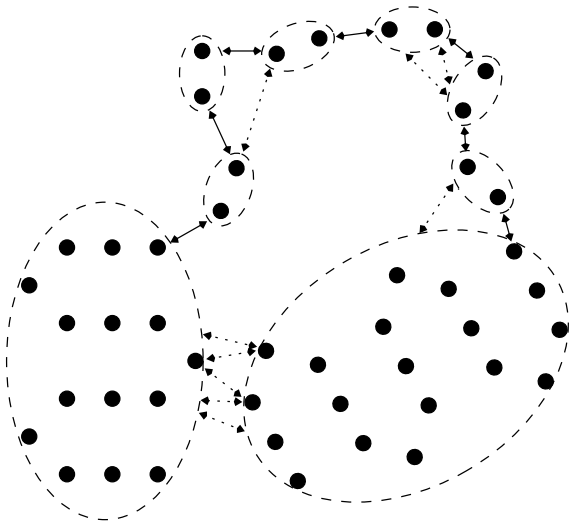


Figure 1.7: Clustered Molecular Dynamics: Atoms (drawn as black spheres) are clustered into rigid objects (dashed ellipses). The forces that the atoms exert onto each other (black arrows suggest attractive, dashed arrows repulsive forces) imply forces and moments acting on the objects. The system can be simulated by tracking the objects' positions.

This *clustered molecular dynamics (CMD)* technique does not add additional realism to the model. Nevertheless it proves helpful in two ways: First, the reduction of parameters speeds up the computational process dramatically. Second, the clustering fits very well into the concepts of a modelling biochemist, who does not think in terms of atoms but of higher level structures. As shown in detail in chapters 5 and 6, it proves especially helpful to be able to simulate objects that are very different in size, say one to several hundred atoms, in order to simultaneously treat multiple levels of resolution. The method is illustrated in figure 1.7.

The aim of section 1.5 is to define the physical properties of a system in which groups of atoms can be clustered into rigid objects and find methods to simulate such systems. The approach only makes sense if the "right" atoms are clustered. For the ribosome, this is discussed in detail in chapter 5.

1.5.1 Rigid Object Kinematics

Instead of tracking single atoms, groups of atoms moving simultaneously will be tracked. Such a group of atoms is called *rigid object*. The characteristic property of a rigid object is that deformations do not occur. In other words: the relative distances and orientations of atoms within the object remain constant.

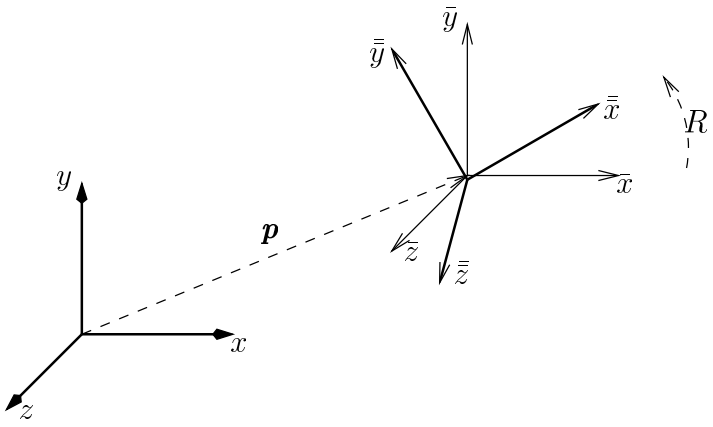


Figure 1.8: Frames used to specify the position and orientation of a rigid body. Inertial frame (denoted by x, y, z), translated frame (denoted by $\bar{x}, \bar{y}, \bar{z}$), and the body frame, which is translated and rotated (denoted by $\bar{\bar{x}}, \bar{\bar{y}}, \bar{\bar{z}}$). Each point a has coordinates $\boldsymbol{a}$, $\bar{\boldsymbol{a}}$, and $\bar{\bar{\boldsymbol{a}}}$ respectively in these frames.

Frames

Mathematically the situation can be handled by introducing a set of constraints, requiring these intra-object relations to be constant. Geometrically this corresponds to introducing a mobile coordinate system, called *body frame*, for each object and keeping the objects' atoms fixed with respect to it.

The body frame has a certain position and orientation with respect to a fixed coordinate system or *inertial frame*. The position is specified by a translation vector $\boldsymbol{p}$. The orientation can be represented in several ways. One way is to use a rotation matrix R. A point a in the body frame has the coordinates $\bar{\bar{\boldsymbol{a}}}$, in the translated frame has the coordinates $\bar{\boldsymbol{a}} = R\bar{\bar{\boldsymbol{a}}}$, and in the inertial frame has the coordinates $\boldsymbol{a}$:

$$\boldsymbol{a} = \boldsymbol{p} + \bar{\boldsymbol{a}} = \boldsymbol{p} + R\bar{\bar{\boldsymbol{a}}}, \quad \text{with } \boldsymbol{a}, \bar{\boldsymbol{a}}, \bar{\bar{\boldsymbol{a}}} \in \mathbb{R}^3. \tag{1.55}$$

For any atom a in an object, the coordinates with respect to the body frame are fixed: $\bar{\bar{\boldsymbol{a}}} = const$. Therefore the first and second order time-derivatives of its position $\boldsymbol{a} = \boldsymbol{p} + R\bar{\bar{\boldsymbol{a}}}$ fall into two components:

$$\dot{\boldsymbol{a}} = \dot{\boldsymbol{p}} + \dot{R}\bar{\bar{\boldsymbol{a}}} \tag{1.56}$$

$$\ddot{\boldsymbol{a}} = \ddot{\boldsymbol{p}} + \ddot{R}\bar{\bar{\boldsymbol{a}}} \tag{1.57}$$

Motion of rigid objects can thus be decomposed into a translational component – change of $\boldsymbol{p}$ – and a rotational component – change of R. The derivatives of $\boldsymbol{p}$ are straight-forward:

$$\dot{\boldsymbol{p}} = (\dot{p}_x, \dot{p}_y, \dot{p}_z)^T \tag{1.58}$$

$$\ddot{\boldsymbol{p}} = (\ddot{p}_x, \ddot{p}_y, \ddot{p}_z)^T \tag{1.59}$$

The rotational component needs slightly more consideration. Note that $\dot{\boldsymbol{a}} = \dot{R}\bar{\bar{\boldsymbol{a}}} = \dot{R}R^T\bar{\boldsymbol{a}}$. Differentiating the identity $RR^T = Id$ results in $\dot{R}R^T = -(\dot{R}R^T)^T$,

thus $\dot{R}R^T$ is a skew-symmetric matrix, that is $\dot{R}R^T \in so(3)$. The velocity $\dot{\bar{\boldsymbol{a}}}$ can therefore be written as cross-product (see appendix A.3):

$$\dot{\bar{\boldsymbol{a}}} = \bar{\boldsymbol{\omega}} \times \bar{\boldsymbol{a}}, \tag{1.60}$$

where $\bar{\boldsymbol{\omega}}$ is called *angular velocity*. The (very same) angular velocity vector has coordinates $\bar{\tilde{\boldsymbol{\omega}}}$ in the body frame. The derivative $\dot{\boldsymbol{\omega}}$ of the angular velocity is called *angular acceleration*. The representation of the orientation has an impact on the kinematics and dynamics formulation and thus on the stability of numerical integration. Euler angles are the classical parameterization of rotation matrices. However, this parameterization yields serious disadvantages.

Euler Angles – Drawbacks

Classically a rotation matrix R is parameterized using Euler angles α, β, and γ, specifying three successive rotations (see appendix A.5)

$$R = R_z(\alpha)R_y(\beta)R_z(\gamma). \tag{1.61}$$

By the chain-rule of differentiation the expression $\dot{R}R^T$ then simplifies to:

$$\begin{aligned} \frac{dR}{dt}R^T &= \frac{R_z(\alpha)R_y(\beta)R_z(\gamma)}{dt}R_z(-\gamma)R_y(-\beta)R_z(-\alpha) && (1.62)\\ &= \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\dot{\alpha} + \begin{bmatrix} 0 & 0 & \cos\alpha \\ 0 & 0 & \sin\alpha \\ -\cos\alpha & -\sin\alpha & 0 \end{bmatrix}\dot{\beta} \\ &\quad + \begin{bmatrix} 0 & -\cos\beta & \sin\alpha\sin\beta \\ \cos\beta & 0 & -\cos\alpha\sin\beta \\ -\sin\alpha\sin\beta & \cos\alpha\sin\beta & 0 \end{bmatrix}\dot{\gamma}. && (1.63) \end{aligned}$$

Thus, the angular velocity is

$$\bar{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\sin\alpha & \cos\alpha\sin\beta \\ 0 & \cos\alpha & \sin\alpha\sin\beta \\ 1 & 0 & \cos\beta \end{bmatrix} \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{pmatrix}. \tag{1.64}$$

This latter matrix is singular for $\sin\beta = 0$. Since the inverse of the matrix appears in the equations of motion, the numerical treatment will become unstable whenever $sin\beta$ approaches 0. This problem can be overcome using unit quaternions to specify rotations.

Unit Quaternions

An alternative to using a rotation matrix R is to use a unit quaternion $\tilde{q}$, with which a rotation can be written as (see appendix A.6):

$$\bar{\boldsymbol{a}} = \tilde{q}\,\bar{\tilde{\boldsymbol{a}}}\,\tilde{q}^*. \tag{1.65}$$

A rotation matrix can be computed for a unit quaternion. Thus conversion from unit quaternions into Euler angles is possible, and vice-versa. The parameters that these conversions yield are not unique (both ways), but the underlying rotation remains the same.

Differentiating $\bar{\boldsymbol{a}}$, with $\bar{\bar{\boldsymbol{a}}}$ constant, yields

$$\dot{\bar{\boldsymbol{a}}} = \frac{d(\tilde{q}\bar{\bar{\boldsymbol{a}}}\tilde{q}^*)}{dt} = \dot{\tilde{q}}\bar{\bar{\boldsymbol{a}}}\tilde{q}^* + \tilde{q}\bar{\bar{\boldsymbol{a}}}\dot{\tilde{q}}^*, \tag{1.66}$$

where $d\tilde{q}^*/dt = (d\tilde{q}/dt)^* = \dot{\tilde{q}}^*$. Inserting $\bar{\bar{\boldsymbol{a}}} = \tilde{q}^*\,\bar{\boldsymbol{a}}\tilde{q}$ into (1.66) results in

$$\dot{\bar{\boldsymbol{a}}} = \dot{\tilde{q}}\tilde{q}^*\,\bar{\boldsymbol{a}}\tilde{q}\tilde{q}^* + \tilde{q}\tilde{q}^*\,\bar{\boldsymbol{a}}\tilde{q}\dot{\tilde{q}}^* = \dot{\tilde{q}}\tilde{q}^*\,\bar{\boldsymbol{a}} + \bar{\boldsymbol{a}}\tilde{q}\dot{\tilde{q}}^*. \tag{1.67}$$

Note that $\tilde{u} = \dot{\tilde{q}}\tilde{q}^* = -\tilde{q}\dot{\tilde{q}}^*$ is a pure quaternion ($\bar{\boldsymbol{u}} = \tilde{u}$) to obtain

$$\dot{\bar{\boldsymbol{a}}} = \dot{\tilde{q}}\tilde{q}^*\,\bar{\boldsymbol{a}} - \bar{\boldsymbol{a}}\dot{\tilde{q}}\tilde{q}^* = \bar{\boldsymbol{u}} \times \bar{\boldsymbol{a}} - \bar{\boldsymbol{a}} \times \bar{\boldsymbol{u}} = 2\bar{\boldsymbol{u}} \times \bar{\boldsymbol{a}}. \tag{1.68}$$

Thus the angular velocity is related to quaternion velocity by the relation $\bar{\boldsymbol{\omega}} = 2\bar{\boldsymbol{u}}$. The angular velocity in the body frame is $\bar{\bar{\boldsymbol{\omega}}} = \tilde{q}^*\bar{\boldsymbol{\omega}}\tilde{q} = 2\tilde{q}^*\dot{\tilde{q}}$.

The quaternion accelerations are related to the angular velocity by

$$\dot{\bar{\bar{\boldsymbol{\omega}}}} = d(\tilde{q}^*\dot{\tilde{q}})/dt = \dot{\tilde{q}}^*\dot{\tilde{q}} + \tilde{q}^*\ddot{\tilde{q}}. \tag{1.69}$$

Premultiply by $\tilde{q}$ to get

$$2\ddot{\tilde{q}} = \tilde{q}\dot{\bar{\bar{\boldsymbol{\omega}}}} - 2\tilde{q}\dot{\tilde{q}}^*\dot{\tilde{q}}. \tag{1.70}$$

This equation nicely relates angular acceleration to quaternion parameterization. It will be used after the relations between acting forces and angular acceleration are discussed.

Object Parameters

A rigid object o is thus completely characterized by a set of atoms $A = \{a_i : \text{atom } a_i \in \text{ object } o\}$, a position vector $\boldsymbol{p}$, and an orientation, specified by a unit quaternion $\tilde{q}$. The notation $o = (A, \boldsymbol{p}, \tilde{q})$ is used to state this.

1.5.2 Rigid Object Dynamics

Force and Acceleration

The aim of this section is to relate the forces acting on an object to the resulting acceleration of the translation and rotation parameters of the object.

In the sequel the position vector $\boldsymbol{p}$ is chosen such that it points to the object's centre of mass:

$$\boldsymbol{p} = \frac{\sum_{a_i \in A} a_{i,m}\boldsymbol{a}_i}{\sum_{a_i \in A} a_{i,m}} = \frac{\sum_{a_i \in A} a_{i,m}\boldsymbol{a}_i}{m} \qquad \text{with the object's mass} \quad m = \sum_{a_i \in A} a_{i,m}. \tag{1.71}$$

According to Newton's second law, the force $\boldsymbol{F}_i$ acting on an atom a_i is related to the atoms acceleration by $\boldsymbol{F}_i = a_{i,m}\ddot{\boldsymbol{a}}_i$. Summing the forces acting on all atoms within an object yields:

$$\sum_{a_i \in A} \boldsymbol{F}_i = \sum_{a_i \in A} a_{i,m}\ddot{\boldsymbol{a}}_i = m\ddot{\boldsymbol{p}}. \tag{1.72}$$

This relation is called the *law of motion of the centre of mass*. Note that for two atoms $a_i, a_j \in A$ within an object the force that atom a_i exerts on atom a_j is the negative of the force that atom a_j exerts on atom a_i according to Newton's third law. Thus the two forces cancel in (1.72), and only external forces matter.

Again, for the rotation a few intermediate steps are necessary to relate *torque* with angular acceleration.

Angular Momentum

Note that the vector quantity

$$\bar{\boldsymbol{L}} = \sum_{a_i \in A} \bar{\boldsymbol{a}}_i \times \dot{\bar{\boldsymbol{a}}}_i a_{i,m} = \sum_{a_i \in A} \bar{\boldsymbol{a}}_i \times (\bar{\boldsymbol{\omega}} \times \bar{\boldsymbol{a}}_i) a_{i,m} \tag{1.73}$$

which is referred to as the *angular momentum* of an object can be written as matrix multiplication:

$$\bar{\boldsymbol{L}} = \bar{I}\bar{\boldsymbol{\omega}}, \quad \text{with} \quad \bar{I} = \begin{bmatrix} \bar{I}_{xx} & -\bar{I}_{xy} & -\bar{I}_{xz} \\ -\bar{I}_{yx} & \bar{I}_{yy} & -\bar{I}_{yz} \\ -\bar{I}_{zx} & -\bar{I}_{zy} & \bar{I}_{zz} \end{bmatrix}, \tag{1.74}$$

where the inertia matrix $\bar{I}$ is defined by

$$\begin{aligned} \bar{I}_{xx} &= \sum_{\bar{a}_i \in A} (\bar{a}_{i,y}^2 + \bar{a}_{i,z}^2) a_{i,m} & \bar{I}_{xy} &= \bar{I}_{yx} = \sum_{\bar{a}_i \in A} \bar{a}_{i,x}\bar{a}_{i,y}\, a_{i,m} \\ \bar{I}_{yy} &= \sum_{\bar{a}_i \in A} (\bar{a}_{i,x}^2 + \bar{a}_{i,z}^2) a_{i,m} & \bar{I}_{xz} &= \bar{I}_{zx} = \sum_{\bar{a}_i \in A} \bar{a}_{i,x}\bar{a}_{i,z}\, a_{i,m} \\ \bar{I}_{zz} &= \sum_{\bar{a}_i \in A} (\bar{a}_{i,x}^2 + \bar{a}_{i,y}^2) a_{i,m} & \bar{I}_{yz} &= \bar{I}_{zy} = \sum_{\bar{a}_i \in A} \bar{a}_{i,y}\bar{a}_{i,z}\, a_{i,m}. \end{aligned} \tag{1.75}$$

Since $\bar{I}$ is symmetric, it can be diagonalized.

The construction can be formulated in body frame coordinates analogously, where

$$\bar{\bar{\boldsymbol{L}}} = \sum_{a_i \in A} \bar{\bar{\boldsymbol{a}}}_i \times \dot{\bar{\bar{\boldsymbol{a}}}}_i a_{i,m} = \bar{\bar{I}}\bar{\bar{\boldsymbol{\omega}}} \tag{1.76}$$

and $\bar{\bar{I}}$ is constant and also diagonalizable. Thus there is a matrix $K \in SO(3)$ with $K^T \bar{\bar{I}} K = \bar{\bar{\bar{I}}}$, where $\bar{\bar{\bar{I}}}$ is a diagonal matrix. Therefore the body-frame can always be attached to the object in such a way that the inertia matrix $\bar{\bar{I}}$ is diagonal. The equivalent statement in mechanics is: the object is aligned with its principal axes of inertia.

The matrix K can be found numerically by eigenvector analysis, e.g with the Jacobi method [PTVF97]. In order not to complicate notation with any additional bars, in the sequel the body frame is assumed to be chosen such that $\bar{\bar{I}}$ is diagonal.

Angular Momentum and Torque

Differentiating $\bar{\boldsymbol{L}}$ reveals that *torque* $\bar{\boldsymbol{M}}$ equals the rate of change of angular momentum

$$\dot{\bar{\boldsymbol{L}}} = \sum_{a_i \in A} \left(\dot{\bar{\boldsymbol{a}}}_i \times \dot{\bar{\boldsymbol{a}}}_i + \bar{\boldsymbol{a}}_i \times \ddot{\bar{\boldsymbol{a}}}_i \right) a_{i,m} = \sum_{a_i \in A} \bar{\boldsymbol{a}}_i \times \ddot{\bar{\boldsymbol{a}}}_i a_{i,m} = \sum_{a_i \in A} \bar{\boldsymbol{a}}_i \times \bar{\boldsymbol{F}}_i = \bar{\boldsymbol{M}}. \quad (1.77)$$

Change of Angular Momentum and Angular Velocity

Differentiating $\bar{\bar{\boldsymbol{L}}} = \bar{\bar{I}} \bar{\boldsymbol{\omega}}$, and using the fact that the inertia matrix $\bar{\bar{I}}$ is constant in the body frame, one obtains:

$$\frac{d\bar{\bar{\boldsymbol{L}}}}{dt} = \bar{\bar{I}} \frac{d\bar{\boldsymbol{\omega}}}{dt}. \quad (1.78)$$

On the other hand $\bar{\bar{\boldsymbol{L}}} = R^T \bar{\boldsymbol{L}}$, which yields

$$\frac{d\bar{\bar{\boldsymbol{L}}}}{dt} = \frac{d(R^T \bar{\boldsymbol{L}})}{dt} = \frac{dR^T}{dt} \bar{\boldsymbol{L}} + R^T \frac{d\bar{\boldsymbol{L}}}{dt}. \quad (1.79)$$

Combine (1.78) and (1.79) to obtain

$$\bar{\bar{I}} \frac{d\bar{\boldsymbol{\omega}}}{dt} = \frac{dR^T}{dt} \bar{\boldsymbol{L}} + R^T \frac{\bar{\boldsymbol{L}}}{dt}. \quad (1.80)$$

Both terms on the right side are known and, since $\bar{\bar{I}}$ is diagonal, (1.80) is easily solved for $d\bar{\boldsymbol{\omega}}/dt$. Repeating steps (1.79) and (1.80) in quaternion notation becomes

$$\bar{\bar{I}} \dot{\bar{\boldsymbol{\omega}}} = \frac{d\bar{\bar{\boldsymbol{L}}}}{dt} = \frac{d(\tilde{q} \bar{\boldsymbol{L}} \tilde{q}^*)}{dt} = \dot{\tilde{q}} \bar{\boldsymbol{L}} \tilde{q}^* + \tilde{q} \dot{\bar{\boldsymbol{L}}} \tilde{q}^* + \tilde{q} \bar{\boldsymbol{L}} \dot{\tilde{q}}^*, \quad (1.81)$$

which, again, can be solved for change of angular velocity in the body frame $\dot{\bar{\boldsymbol{\omega}}}$.

Inserting the result into (1.70) yields the equations of motion, parameterized with quaternions, which can be integrated numerically using the methods discussed in section 1.4.2.

Conclusion

The complete system consists of a multitude of atoms. Each atom belongs to a rigid object. Tracking the system requires the forces to be evaluated which the atoms exert on each other. The forces acting on the atoms imply forces and moments acting on the rigid objects.

The motion of a rigid object can be decomposed into a translational and a rotational part. The translational acceleration solely depends on the sum of forces $\boldsymbol{F} = \sum_{a_i \in A} \boldsymbol{F}_i$ acting on the object, as described by (1.72). The rotational acceleration is related to the torque $\bar{\boldsymbol{M}} = \sum_{a_i \in A} \bar{\boldsymbol{a}}_i \times \bar{\boldsymbol{F}}_i$ acting on the object by relations (1.77) and (1.81).

When each atom was allowed to move individually, the differential equation $\ddot{\boldsymbol{a}}_i = f_i(\boldsymbol{s})/a_{i,m}$ in (1.37) related the motion of the atoms in the system with the

underlying energy potential. Now the motion within the system can be reformulated in terms of object positions and orientations by

$$\ddot{\boldsymbol{p}}_i = pf_i(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{no}, \tilde{q}_1, \ldots, \tilde{q}_{no}) \quad (1.82)$$

$$\ddot{\tilde{q}}_i = qf_i(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{no}, \tilde{q}_1, \ldots, \tilde{q}_{no}, \dot{\tilde{q}}_1, \ldots, \dot{\tilde{q}}_{no}) \quad (1.83)$$

where no is the number of objects in the system, $\boldsymbol{p}_i$ is the position of object o_i, $\tilde{q}_i$ is its orientation, and $pf_i(), qf_i()$ are functions that state the relations discussed above in this section.

Instead of integrating each atom's position numerically, the positions $\boldsymbol{p}$ and orientations $\tilde{q}$ of the rigid objects are integrated. This can be done with the numerical integration schemes discussed in section 1.4.2.

1.6 Clustered Molecular Dynamics – a Minimization Tool

Molecular dynamics models a system of atoms. It is generally used to simulate that system, to track it for a certain time – and often to find a stable conformation of the atoms. The difference is: in simulation, conservation of energy is essential. Energy is often used to double check that the simulation of the system is functioning properly. On the other hand, energy minimization, which is used to find stable conformations, reduces the total energy of the system.

1.6.1 Principle

Example (a)

As a simple example, consider a point mass in a parabola subjected to a gravitational field, as depicted in figure 1.9 a. Neglecting friction, the mass is drawn down and moves constrained by the parabola. The potential energy of the mass is proportional to the square of its height - thus a plot of the energy will follow the contour in its ups and downs. Assume a zero initial velocity and release the mass at a certain height as shown in the figure. It glides down through the middle, up again on the other side, finally returning to its original position, where the motion starts anew. If the system is stopped at a random time, the position of the mass is arbitrarily somewhere in between the starting point and its counterpoint of equal height.

At the middle position, the mass has minimal potential energy. How can this position be found? The answer is to track the system while continuously removing energy. There are many possibilities for removing energy from the system, and two will be discussed in the following:

One way to remove energy is to introduce a friction force, proportional to the mass's velocity but opposite in direction. The point mass now glides smoothly down the parabola and then up on the other side to a reduced height.

A less intuitive way to remove energy, that turned out to be highly efficient in application, is to stop the system from time to time, remove kinetic energy (i.e.

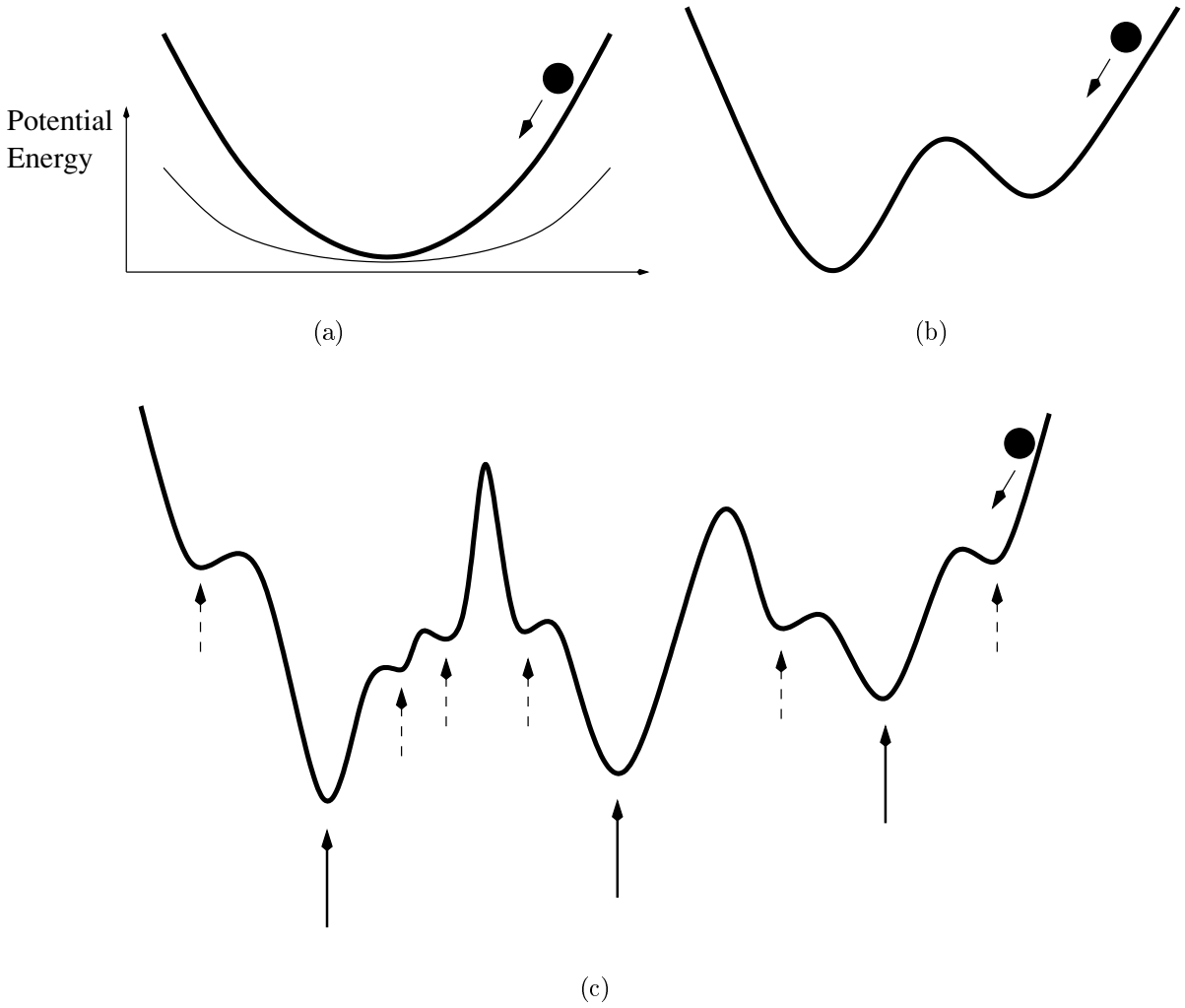


Figure 1.9: Stable conformations, as described in section 1.6.1

set all velocities to zero), and restart. In the point-mass-parabola-system this corresponds to catching the mass after a certain time interval and releasing it at the same position, but without an impulse. By constantly decreasing the catch time interval, the minimum can be reached quite rapidly.

Example (b)

A slightly more complicated example is depicted in figure 1.9 b. Instead of a parabola, the constraint now has two local minima. Both of them are *stable conformations*; if put there without an initial velocity, the mass will rest there.

Neither energy removal method can guarantee that the mass will end in the lower – global – minimum. However, both guarantee that it ends in a local minimum.

Example (c)

The third example, figure 1.9 c, raises a new question: Is any local minimum equally acceptable? Let the point mass rest in one of the shallow minima (those labelled with a dashed arrow in the figure), and subject the system to a minor earthquake. There is a reasonable chance that the mass will "hop out" and fall into one of the lower minima (labelled with non-dashed arrows). There it will rest, even if subjected to continuous seismic activity. Two conclusions from this are that sufficiently deep local minima are preferred stable conformations, and that introducing some external energy can help to find them.

1.6.2 Transfer to the Molecular Model

The energy potential (1.12), defined in section 1.2, corresponds to a multidimensional parabola in the previous examples, the vector 1.33 containing all atom positions corresponds to the position of point mass then and the same general principle applies: energy can be minimized by tracking the system and continuously removing energy.

Energy Removal

As suggested in the examples, energy removal is accomplished in two ways: First, a friction force is introduced for each atom. The direction of the force is opposite to the atoms velocity vector, the magnitude is lineary proportional to the atoms speed. Second, the system is stopped frequently and restarted either with zero velocities, or with randomized velocities. The latter corresponds to the seismic activity of previous example c and prevents the system from getting stuck in shallow local minima.

Both methods do not actually simulate any physical behaviour of atomic systems but are introduced to quickly reduce the energy within the system.

Limiting the Energy Functions

Additional refinement of the method is necessary, because the potential (1.12) is slightly more complicated than suggested by the examples.

Due to the Lennard-Jones function (1.6), the slopes of the potential (1.12) can be very steep, which causes high accelerations and thus high velocities. High velocities

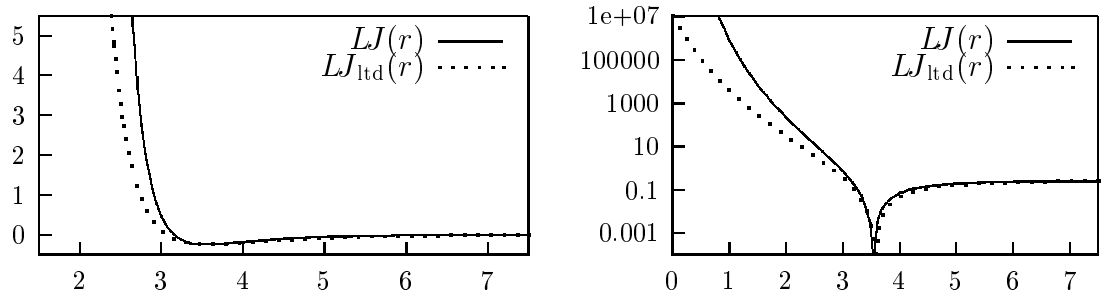


Figure 1.10: Lennard-Jones and limited Lennard-Jones function. The parameters are $A = 10^6 \left[\frac{\text{kcal}\cdot\text{Å}^{12}}{\text{mole}}\right]$, $B = 10^3 \left[\frac{\text{kcal}\cdot\text{Å}^{6}}{\text{mole}}\right]$, and $LJ_{\max} = 10^7$. Energy in kcal/mole is plotted on the y-axis, distance in Ångstrom on the x-axis.

present a problem for the numerical integration scheme. Limiting the Lennard-Jones potential experimentally proved to be helpful in this situation, especially during start-up of the minimization process (see also section 6.4).

The standard Lennard-Jones potential was defined in (1.6) to be

$$LJ(r) = \frac{A}{r^{12}} - \frac{B}{r^6}. \tag{1.84}$$

It can be limited to a maximum $LJ_{\max}$ by slightly shifting the parameter, while requiring the function minimum to keep its position and value:

$$LJ_{\text{ltd}}(r) = \frac{A_{\text{ltd}}}{(r+s)^{12}} - \frac{B_{\text{ltd}}}{(r+s)^6} \tag{1.85}$$

with

$$LJ_{\text{ltd}}(0) = LJ_{\max} \tag{1.86}$$

$$LJ_{\text{ltd}}(r_0) = LJ(r_0) \tag{1.87}$$

$$\frac{d\,LJ_{\text{ltd}}}{dr}(r_0) = 0 = \frac{d\,LJ}{dr}(r_0). \tag{1.88}$$

This can be obtained by choosing A_{ltd} and B_{ltd} as follows:

$$r_0 = \sqrt[6]{\frac{2A}{B}} \tag{1.89}$$

$$\epsilon = \frac{B^2}{4A} \tag{1.90}$$

$$s = \frac{r_0}{\sqrt[6]{1+\sqrt{1+\frac{LJ_{\max}}{\epsilon}}} - 1} \tag{1.91}$$

$$A_{\text{ltd}} = \epsilon(r_0+s)^{12} \tag{1.92}$$

$$B_{\text{ltd}} = 2\epsilon(r_0+s)^6. \tag{1.93}$$

This result is derived by parameterizing the Lennard-Jones function into $LJ(r) = 4\epsilon\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right)$, parameterizing its limited version analogously, and remarking that ϵ is the negative of the minimum function value $LJ(r_0)$.

Adaptive Timestep Integration

The shifted Lennard-Jones Potential (1.85) in practice limits the force operating on a single atom. This allows the clustered molecular dynamics minimization to be run with an adaptive timestep integration as described in section 1.4.2.

The timestep is controlled by computing maximum forces and torques acting on the objects at the current time, and ensuring a certain maximum object movement per timestep.

1.7 Software Engineering

The section gives an overview of the clustered molecular dynamics software package that was implemented during this work at the *Max-Planck-Institut für molekulare Genetik* for application to the *E.coli* ribosome (cf. part II).

1.7.1 Package Overview and Summary

The aim of the development was a multi-scale molecular mechanics package for biochemical applications, with the capability of incorporating data of different types and resolutions from a variety of sources (cf. chapter 4). These data result in various demands that are expressed as energy functions. At the atomic level stereochemical demands (cf. section 1.2.1, 1.2.2) as well as Van-der-Waals effects (cf. section 1.2.3) are included. At a coarser scale, distance (cf. section 1.2.4) and shape (cf. section 1.2.5) constraints are included.

In order to handle potentially contradictory data and not to destroy known building blocks of the structure, the clustering technique as formally derived in section 1.5 was implemented, allowing arbitrary sets of atoms to be clustered into rigid objects and to be treated computationally as such.

The problems to be solved with the package are refinement and evaluation of molecular models.

1.7.2 Quality Attributes

Standard IEEE quality attributes are listed here and their importance for the package is evaluated in order to justify the design choices made [IEE83, Fai85].

Requirement	Ranking
Accuracy	10
Portability	6
Reliability	10
Efficiency	9
Robustness	1
Correctness	10

Table 1.1: Quality attributes according to [IEE83] and respective ranking on a scale from $1-10$.

Portability

Portability, being the ease with which software can be transferred from one computer system to another, must be assured for this molecular mechanics package in so far as target machines are medium to high-end workstations, and ports to newer, faster

versions of these workstations must be feasible. Vector-computers and highly parallel machines are not accessible for most modelling teams, therefore portability to such machines was ranked with low priority.

Efficiency

From the molecular modeller's point of view, efficiency – the extent to which software performs its intended functions with a minimum consumption of computer resources and computing time – has two aspects: efficiency of model evaluation and efficiency of the refinement process.

Evaluation of the model is done frequently during the interactive fitting process (cf. section 6.3.2) and thus is required to be feasible in limited time. A response time of one minute for checking the validity of a model with 200.000 atoms (cf. part II) is considered acceptable by biochemists modelling in our institute.

Refinement of the model (cf. section 6.4) is much more time consuming and not an interactive task. Considering the size of the model to be refined, for the problem to be solvable at all, algorithms need to be designed efficiently.

Robustness

As well as the input to the modelling package, with a plenitude of parameters to be set, the interpretation of its output requires understanding of the computational process. Therefore robustness, being the extent to which software can continue to operate correctly despite the introduction of invalid inputs, is not an issue for this package.

Reliability, Accuracy, and Correctness

Reliability of the program package – the ability to perform the required functions under stated conditions for a stated period of time –, accuracy – the assessment of freedom of error –, and correctness – the extent to which the software is free from design and coding effects – are requirements inherent to scientific software. Correctness of the computed results is important for the biochemist to rely on the method.

All parameters are specified before the computation, and no user or system requests need to be handled during the computational process. Careful derivation of the molecular mechanics mathematics was done in sections 1.4 and 1.5. For the corresponding implementation input-output assertions are used. On the other hand, a molecular mechanics package can not become directly dangerous to human life. Therefore a formal proof of correctness, e.g. with the weakest preconditions method, is not attempted.

1.7.3 Software Requirements Specification

Development and Operating Environments

As mentioned in the previous section, development and operating target machines are medium to high-end workstations. For the actual development a restriction to

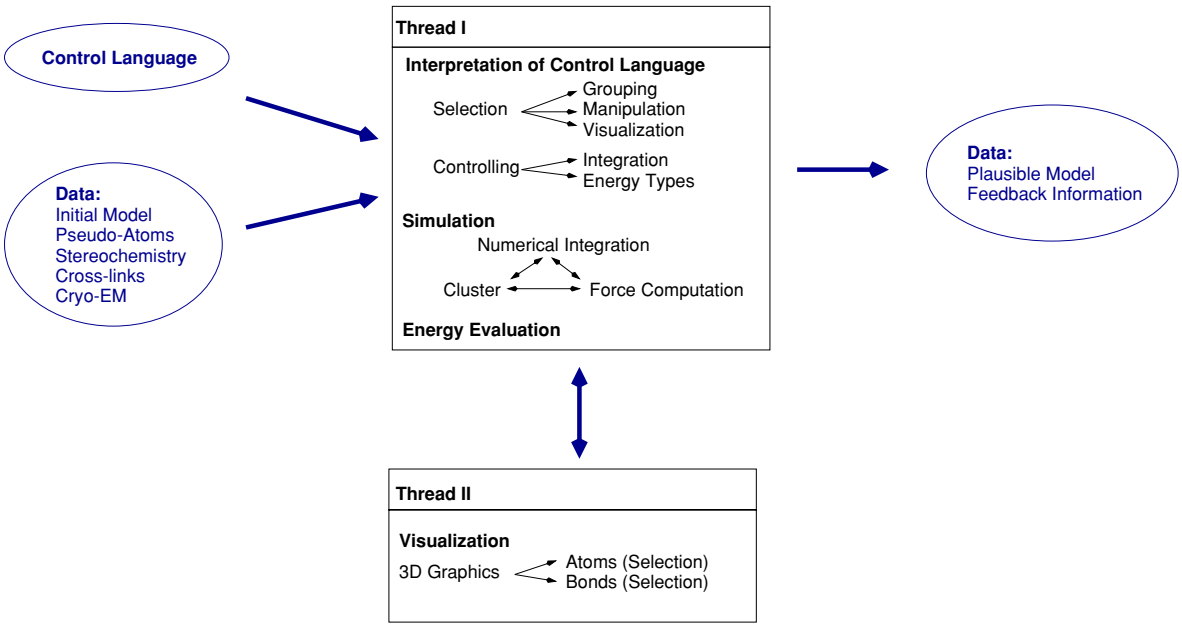


Figure 1.11: Data flow within the molecular mechanics package.

available platforms was made: Silicon Graphics machines running Irix, Dec/Alpha machines running OSF, and PCs running Linux. Portability between these computer systems was established using standard languages (C++), library functions, and operating system file functions.

External Interfaces and Data Flow

The data flow is schematically depicted in figure 1.11. Input to the computations are a molecular model, stereochemical descriptions, density data, cross-linking data, conrol-, clustering- and pseudo-atom information.

Molecular model files use the protein data base (PDB)[1] standard. This widely used file-format for molecules allows each atom to be assigned a specific position and also enables some higher level structural properties to be stored, such as sequences, nucleotides, and helices (cf. chapter 3).

Stereochemical descriptions are specified using nucleotide patterns (cf. chapter 5, the descriptions are listed in appendix B).

Density data are specified per voxel, adjusted with the molecular coordinates before computation, and stored byte-wise in a file (cf. section 5.5).

For the specification of the cross-linking data (cf. section 5.4), a simple selection mechanism, specifying the atoms involved, is used. Each cross-link consists of two such selections, combined with the specification of a desired length and a force constant. Arbitrary cross-links, sorted according to biochemical reasoning, are deposited in several files to be incorporated (see appendix C).

A similar selection mechanism is used for the control files, clustering files and pseudo-atom process-description files. These describe which models to evaluate or

[1] http://www.rcsb.org/pdb/

refine, what groups to cluster at what time of the process, and which pseudo-atoms to attach in the respective stages (cf. section 6.4).

Output of the molecular mechanics package are feedback information and/or a refined molecular model. Feedback information (cf. section 6.3.2) is textual, i.e. a list of problems, which can later be visualized if necessary.

Functional Requirements and Acceptance Criteria

Contrary to classical algorithmic design, where a fixed and generic problem definition requires an exact solution, in bioinformatics a biologically relevant solution is desired, which cannot be coached in terms of exact optima, because constraints and cost functions are heuristic. Therefore experimental validation takes priority [Len98].

Nevertheless, in cooperation with the modelling biochemist, tolerances for the individual demands to be incorporated can be defined. In an ad hoc approach for this work the following was required: After computational refinement, the model has no Van-der-Waals overlaps, bond length divergence is no more then 0.4 Å from optimal, bond angle divergence is no more than 30 degrees, density tolerance is 13 Å. No unique acceptance measure was defined for cross-links for their different methods of generation and biochemical relevance (cf. section 4.2.5).

Much more important for acceptance than these requirement specifications were model walkthroughs with biochemists, inspecting arbitrary and critical parts of refined particles for chemical plausibility.

Foreseeable Modifications and Enhancements

Visualization and interactive manipulation extensions (cf. chapter 2) were integrated. The graphical interface was established using the OpenGL/Motif combination, which is available on the operating platforms mentioned above.

The visualization incorporates the display of the complete particle with zooms to parts where potential problems arise. Interactive manipulation, i.e. shifting and rotation of clusters, can then be used to deal with problems.

1.7.4 Software Design

The package is implemented in C++, consists of 8000 lines of code, and follows an object oriented paradigm facilitating later extension. The individual classes are discussed in the following paragraphs, the interaction between classes is sketched in figure 1.12.

`Control Class`

A high-level process description language was designed for the computational processes to be controlled. This language controls basic input/output features and the operating of simulation and minimization runs; it allows boolean selection of specific groups of atoms (cf. section 2.2). These selections can then be clustered into rigid objects or can be spatially fixed. Furthermore the language allows the attachment

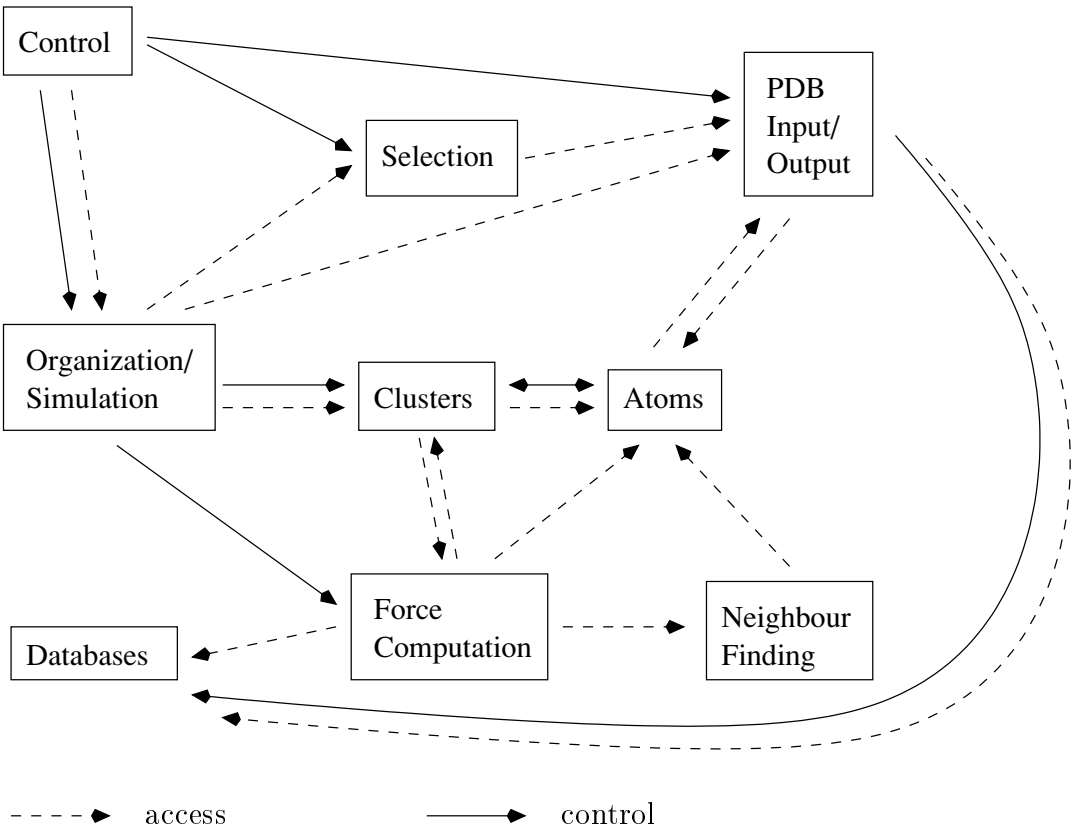


Figure 1.12: Class interaction within the package.

of pseudo-atoms to rigid objects and the control of which atoms are included into the computational processes.

The internal mechanism of boolean selection was implemented into a separate `selection class`.

`Database Classes`

Nucleotide patterns (cf. section 5.1) are stored in a separate database, as are cross-linking data (cf. section 5.4) and data reflecting the cryo-electron microscopy density (cf. section 5.5).

These databases are read with program startup. The nucleotide patterns are used to bind the corresponding atoms when a PDB file is read, and to establish the corresponding `bond length` and `bond angle` force functions. `Cross-linking` and `electron-microscopy density` data are accessed during the computation of forces and for the evaluation of the corresponding energy.

`PDB Input and Output Class`

This class reads and writes PDB files. When reading, each atom is identified with a type and position. Force functions for the particle read are then initialized according to the databases. Before writing, atom positions are updated from the `atom class`.

A write-only mechanism was built for Inventor/VRML files, allowing the state

of the system to be visually controlled with external programs. This was used in particular for controlling the placement of pseudo-atoms relative to nucleotides and RNA while defining the ribosome specific energy potential (cf. chapter 5).

`Organization / Simulation Class`

This class implements organizational functionality required by the `control class` such as clustering all atoms of a selection or attaching pseudo atoms of a special kind to all clusters in question. More importantly, it is in charge of performing the actual energy evaluation and energy minimization.

For the energy minimization an Adams-Bashford-Moulton predictor-corrector algorithm (cf. section 1.4.2) was used for integrating the objects' positions and orientations over time. Additionally, an adaptive integration (cf. section 1.4.2) was implemented, which adjusts the timestep according to the maximal forces and moments currently occurring in the system.

`Energy / Force Calculation`

There are currently five types of energy/force function classes: `bond`, `bond angle`, `Van-der-Waals`, `cross-linking`, and `density energy`. Additional energy functions can be implemented as separate classes and integrated into the package.

Energy evaluation is based on section 1.2, force calculation is based on the forces, as derived in section 1.3. The individual forces act on the `atoms` and thus on the `clusters`. Searching neighbours for potential Van-der-Waals interactions is based on a cell subdivision algorithm [Rap95] listed as `neighbour finding class` in figure 1.12 above.

Within rigid objects only those atom positions are evaluated that are explicitly necessary for calculating energy functions. Furthermore, unnecessary intra-object energy functions are not evaluated.

`Cluster Classes`

Any number of atoms can be clustered into a rigid object. Hierarchical clustering, i.e. the clustering of several rigid objects into a new rigid object, is possible. Three subclasses are to be distinguished: `Single atom clusters`, `multiple atom clusters`, and `hierarchically grouped clusters`.

The orientation of individual atoms is irrelevant for the energy potential. Therefore `single atoms` are treated as objects with positional parameters only. `Multiple atom clusters` have positional and rotational parameters. For each such cluster, the positional parameters are represented as a three-dimensional vector, the rotational parameters as a unit quaternion. `Hierarchical clusters` – clusters consisting of other clusters – are derived from `multiple atom clusters` and keep additional information on the cluster objects they contain.

Pseudo-atoms (cf. sections 5.3.1 and 5.3.2) can be attached to any `multiple atom cluster`.

`Atom Class`

This class holds the atoms' type and its absolute position as a three-dimensional vector of floating point numbers. When the atom is clustered, it also holds the relative position of the atom within the rigid object. Among other things it is responsible for informing the clusters of the forces acting on the atoms within the clusters. `Pseudo-atoms` are atoms with an additional radius specification.

1.7.5 Implementation

An algorithmic description of the simulation classes of the clustered molecular dynamics software package is given in this section. The focus is on the core functionality; input/output, the parsing and interpretation of the command language, graphics, setup, connectivity, neighbour-finding, energy evaluation, and database issues are not addressed. The somewhat complex organization of different cluster classes, that use a common virtual base class, is simplified in this description for readability to just one multiple atom cluster class.

Class 1: Control

control :: main
start graphics thread (optional)
start interpretation control loop

control :: start graphics thread
initialize data
initialize and realize widgets
start Xt main loop

control :: start interpretation control loop
repeat
parse command and perform specified actions
until end command issued

Class 2: Simulation

member variables
list of clusters
list of force functions
ΔT integration timestep

simulation :: clustered molecular dynamics
{ setup cluster objects }
for all clusters c **do**
c → compute centre of mass
c → setup local coordinates
c → clear predictor-corrector history
{ integration loop }
for number of timesteps **do**
initialize neighbourfinding
compute all forces
adaptive timestep integration control
for all clusters c **do**
c → compute acceleration
c → predictor step
compute all forces
for all clusters c **do**
c → compute acceleration
c → corrector step
evaluate energy
inform graphics thread of need to update

simulation :: compute all forces
{compute Van-der-Waals forces according to spatial arrangement of atoms}
for all Van-der-Waals forces F occurring **do**
for both atoms a involved **do**
$a \rightarrow$ force acting (a, F)
{compute other forces according to force function list}
for all other forces F **do**
for all atoms a involved **do**
$a \rightarrow$ force acting (a, F)

simulation :: adaptive timestep integration control
check whether integration timestep ΔT needs to be changed according to the maximum forces, maximum moments, and maximum velocities in the system
if ΔT needs to be changed **then**
compute new ΔT
for all clusters c **do**
c → adjust predictor-corrector history ($\Delta T_{new}, \Delta T_{old}$)

Class 3: Cluster

member variables

p 3D position vector in inertial frame, pointing to the centre of mass
q quaternion
$\dot{p}$ position velocity
$\dot{q}$ quaternion velocity
$\ddot{p}$, $\ddot{p}_{t-\Delta T}$, $\ddot{p}_{t-2\Delta T}$ position acceleration, present and past
$\ddot{q}$, $\ddot{q}_{t-\Delta T}$, $\ddot{q}_{t-2\Delta T}$ quaternion acceleration, present and past
f 3D vector force acting on cluster
m 3D vector torque acting on cluster
list of atoms belonging to the cluster

cluster :: compute centre of mass

$p := \frac{1}{\text{number of atoms in cluster}} \sum_{\text{atoms } a \text{ in cluster}} \text{a} \rightarrow \text{p}$

cluster :: setup local coordinates

compute inertia matrix I, according to equations (1.75)
Jacobi diagonalization: compute orthogonal matrix R such that $R^T I R$ diagonal
for all atoms a in cluster **do**
$a \rightarrow prel := R^T(a \rightarrow p \quad - \quad p)$

cluster :: force acting (F, $apos$)

{ force F is acting at position $apos$ in inertial frame coordinates }
$f := f + F$
$m := m + (apos - p) \times F$

cluster :: compute accelerations

compute $\ddot{p}$ according to equation (1.82)
compute $\ddot{q}$ according to equation (1.83)

cluster :: clear predictor-corrector history

$\ddot{p} := 0, \quad \ddot{p}_{t-\Delta T} := 0, \quad \ddot{p}_{t-2\Delta T} := 0$
$\ddot{q} := 0, \quad \ddot{q}_{t-\Delta T} := 0, \quad \ddot{q}_{t-2\Delta T} := 0$

cluster :: adjust predictor-corrector history $(\Delta T, \Delta T_{old})$

{ called after ΔT_{old} was adjusted to $\Delta T = \Delta T_{new}$ }
compute $\ddot{p}_{t-\Delta T}$, $\ddot{p}_{t-2\Delta T}$ from $\ddot{p}$, $\ddot{p}_{t-\Delta T_{old}}$, $\ddot{p}_{t-2\Delta T_{old}}$ with polynomial interpolation
compute $\ddot{q}_{t-\Delta T}$, $\ddot{q}_{t-2\Delta T}$ from $\ddot{q}$, $\ddot{q}_{t-\Delta T_{old}}$, $\ddot{q}_{t-2\Delta T_{old}}$ with polynomial interpolation

---

Class 3: Cluster (continued)

cluster :: predictor step

compute p_{est}, q_{est} according to equation (1.47)
$p := p_{est}$, $q := q_{est}$
normalize q
compute $\dot{p}_{est}$, $\dot{q}_{est}$ according to equation (1.48)
$\dot{p} := \dot{p}_{est}$
$\dot{q} :=$ closest vector to $\dot{q}_{est}$, which is orthogonal to q

cluster :: corrector step

compute p, q according to equation (1.50)
normalize q
compute $\dot{p}$, $\dot{q}$ according to equation (1.51)
$\dot{q} :=$ closest vector to $\dot{q}$, which is orthogonal to q

Class 4: Atom

member variables

p	3D position vector in inertial frame
$prel$	3D position vector in cluster coordinates
$cluster$	that it belongs to

atom :: force acting (F)

$cluster \rightarrow$ force acting (F, p)

Chapter 2

Interactive Molecular Modelling

The various methods for gaining insight into structure of biological particles provide informations of different type and quality. Some of these can be handled computationally as described in chapter 1, others require biochemical understanding and reasoning. This reasoning can be assisted by visualization techniques. The graphical display of molecular models supports the understanding in general by providing the atomic coordinate positions, allows a direct feedback from energy evaluation to be gained, and is necessary for interactive manipulation.

The aim of this chapter is to outline methods for visualization and interactive manipulation, as well as those for selection of sets of atoms for investigating large molecular systems such as the bacterial ribosome.

2.1 Methods for Visualization

The application of visualization techniques for investigating molecular structures reflects the choice of the degree of abstraction or simplification of the underlying model.

Basic assumptions for visualization discussed in this section are the same as those made in 2.1 for the computational models in the previous chapter: atoms are defined by their position in a coordinate system, and these positions can change with time.

Depending on the problem to be solved, molecular models can be represented in different ways: Often used for all-atom visualization are:

- *wire-representations* where the covalent bonds are drawn as lines (plate 2.1),
- *stick-*, *licorice-*, or *neon-representations* where the covalent bonds are drawn as cylinders possibly with rounded caps (plate 2.1),
- *ball-and-stick-representations* where the bonds are drawn as cylinders and the atoms as spheres (plate 2.1), and
- *CPK*[1]*-representations* where the atoms are drawn as space-filling spheres, which overlap for covalently bound atoms (plate 2.1).

[1] after Corey, Pauling, and Koltun

At a somewhat higher level of abstraction there are representations such as:

- *ribbon-/ tube-representations* where the molecular chain (cf. chapter 3) is drawn as a ribbon or tube (plate 2.1),
- *cylinder-representations* where building blocks sich as helices of ribonucleic acid which are defined in section 3.1.2 are drawn as cylinders, and
- *plate-representations* where carbon rings such as sugar or the bases in nucleic acids (cf. section 3.1.2) are filled (plate 2.1).

Often visualization methods are mixed and combined with colour coding so as to point out functionally important parts or to reflect specific data or energetic relations.

Computer graphics offer elaborated methods for the rendering of these different representations. Their three-dimensional components, spheres, cylinders, polygons, are projected onto a two-dimensional viewport – comparable to taking a snapshot of a virtual scene with a virtual camera [FvDFH90].

For the visualization of density information, either volume rendering techniques can be applied, or the density information is converted into a surface by thresholding (e.g. plate 3.5). Transparency allows simultaneous visualization of the surface of and of the interior of a particle (e.g. plates 6.3).

Dynamic molecule motion can be displayed by drawing trajectories or by rendering successive snapshots taken at different timesteps. Successive snapshots are also rendered for motion within the molecule that is caused by input interaction (see section 2.3).

Limitations of visualization are given by rendering time – the more objects to be displayed, the longer it takes – and by possible visual complexity of structures, which are totally confusing when displayed completely. This can be overcome by selection methods discussed in section 2.2.

Programmes such as Amber[2], Erna3D[3], Molmol[4] Molscript[5], Quanta[6], Rasmol[7], Ribbons[8], and VMD[9] offer most of these features and are often specialized on individual applications.

The plates in this thesis were partially generated with Molmol in combination with the Povray raytracing programme [KBW96], and partially with Erna3D extended with rendering tools based on the Silicon Graphics Inventor package.

2.1.1 Modelling Feedback by Energy Evaluation

When modelling larger particles interactively, many different data need to be incorporated (cf. chapter 4). These are sometimes contradictory and can only be partially reflected by an energy potential function such as (1.12). It is therefore

[2] http://www.amber.ucsf.edu/amber/
[3] http://www.mpimg-berlin-dahlem.mpg.de/~ag_brimacombe/
[4] http://www.mol.biol.ethz.ch/wuthrich/software/molmol/
[5] http://www.avatar.se/molscript/
[6] http://www.msi.com/
[7] http://www.bio.cam.ac.uk/doc/rasmol/
[8] http://www.cmc.uab.edu/ribbons/
[9] http://www.ks.uiuc.edu/Research/vmd/

essential for the modeller to be able to get rapid feedback on how good his model fits the data.

Often only specific energy functions are of interest during a certain modelling stage. For example, the model might not be detailed enough to check the bond or bond angle energies, although Van-der-Waals energies remain important in order to avoid accidental overlaps.

Specification of which energies are to be evaluated is combined with sorting, either by gravity of the problem or by position within the segment.

A textual output of such sorted data can be accompanied by graphical feedback: Colour coding of atoms according to the energies signals regions with cumulated problems. Highlighting specific functions, for example single cross-links (cf. section 4.2.5) to be watched, can be helpful.

2.2 Methods for Selection

Selection of atoms or clusters of atoms within a molecule is relevant for adapting display style and colours as well as for the interactive manipulation that will be discussed in the following section.

The selection process can be realized as a direct picking action, using a mouse or similar input device. For large molecules a selected point on the screen can correspond to many atoms, thus redundancy removal is necessary. A common strategy is to circle through a hit-list, where every successive click on the same screen spot selects another atom which is displayed there.

Another way to pinpoint atoms within a molecule is by use of selection criteria. For this purpose it is helpful to consider the structural organization of biomolecules, which are often organized in strands of repeating groups of atoms. These groups of atoms are referred to as residues. For example, in nucleic acids nucleotides and in proteins amino acids are residues (cf. chapter 3). Strands may be interconnected and form higher level building blocks as for instance helices in nucleic acids (cf. figure 3.4). Standard criteria for selection are helix names, residue types, residue numbers, atom types, atom numbers, and whether atoms are within chains or sidechains. Spatial closeness to specified atoms is another useful criterion. In combination with boolean operators these criteria allow precise and flexible selection of specific sets of atoms.

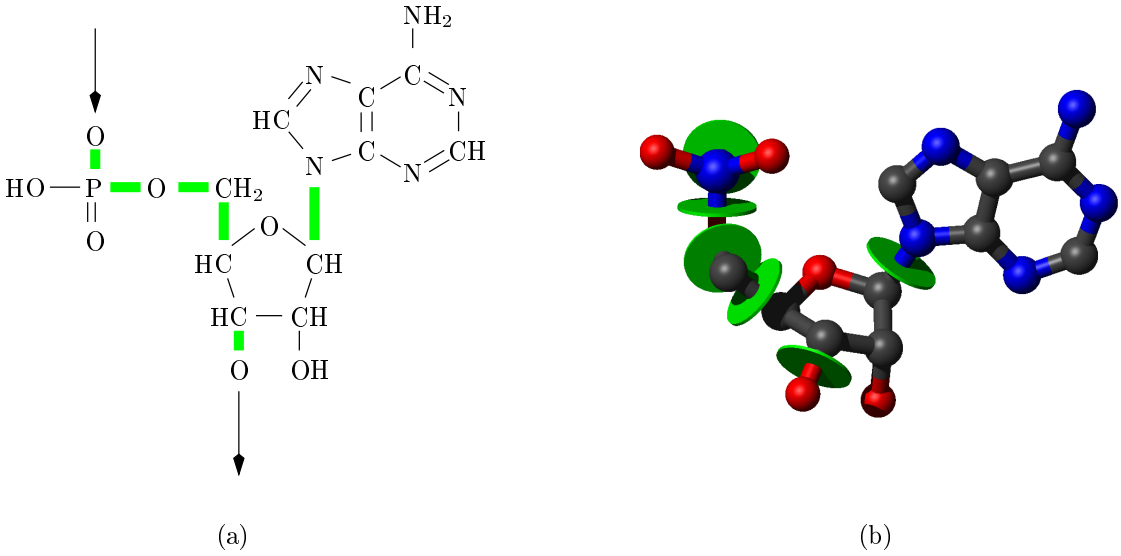
Attributes, such as invisibility and colour can be assigned to the selected atoms. This allows visualization of only the parts of the molecule to be modelled. It also allows the rapid implementation of colouring schemes, for example the following sequence produces the standard colouring used in plate 2.1.

```
select atom.type = "C"   colour grey
select atom.type = "O"   colour red
select atom.type = "N"   colour blue
select atom.type = "H"   colour lightgray
```

In combination with a sequence such as

```
select not residue.number  1 - 30     render invisible
```

only the first thirty residues of a molecule are displayed.



(a) (b)

Figure 2.1: Some bonds within molecular building blocks can be considered as rotation axes. This is depicted here for a nucleotide of ribonucleic acid, where the axes are marked in (a) and correspondingly labelled with a disk in (b).

2.3 Methods for Interactive Manipulation

There are two approaches for the interactive editing of atomic systems: the manipulation of torsion angles and the placement of selected parts of the system.

2.3.1 Torsion Angle Adjustment

Within a molecular strand, some bonds can be considered fixed whereas others can be considered rotation axes with a degree of freedom (figure 2.1). Physicochemically this view is plausible, because the energies for changing the rotation angles of the torsion axes are small [MH87]. The angles can individually be changed, rotating the rest of the strand accordingly, in order to manipulate the molecule. The approach resembles forward kinematic models in robotics, and similar specification schemes can be applied [HH90, MLS94]. For interactive access, every manipulable angle within a building block is assigned to a dial on a dialboard. This technique can be very efficient for modelling short chains with a high degree of precision. For large molecules local manipulations can result in unwanted large scale effects; in order to avoid these, strands can be decomposed into subsegments.

Obviously CPK and ribbon rendering techniques are not suitable for this kind of editing since the axes need to be clearly visible.

2.3.2 Movement of Selected Objects

The other approach for editing molecular systems is the selection of specific groups of atoms, e.g. ribonucleic acid nucleotides or helices (cf. section 3.1.2), with the methods described in section 2.2 to be translated or rotated. Just like in any CAD sys-

tem translations and rotations can be specified by different input devices (keyboard, mouse, spaceball). Such a manipulation can leave the molecule in a stereochemically impossible situation: Covalent bonds are potentially stretched, the concerned atoms may be not even spatially close. Furthermore, bond angles can be imprecise and Van-der-Waals conflicts can occur. These problems can be mastered in several ways:

The human can be made responsible. This can be necessary for large scale movements, but results in immense work in small scale details: adjusting every single atom of a molecular chain by hand is an overwhelming task.

Potential energy functions such as the one defined in section 1.2 can be used to calculate the consequences of the application of a force on a selected atom or on a group of atoms according to section 1.4. This approach is used e.g. by the VMD[10] system. The clustered molecular dynamics approach can be used to keep the structure of a selected group of atoms intact: Instead of applying a force onto every single atom, the selected atoms first are clustered into a rigid object, and then a force or torque is applied to the object.

Approximation methods are computationally less intensive than the potential energy methods mentioned above and therefore often used for realtime applications. Several strategies exist:

The most computationally intensive part of the above potential is the calculation of the Van-der-Waals forces. Removing these from the potential function thus speeds up the calculations dramatically. For the modeller the effect is similar to switching off collision detection in a 3D modelling system. The resulting possibility of shifting of components through other components turns out to be useful at early stages of the modelling process.

Another strategy, used by the programme Erna3D [Mue93, MB97a], is to use a local potential, which describes the structure of the molecular chain(cf. section 3.1.2) around the atoms currently selected for movement. Thus considerably fewer atoms need to be considered as with a complete potential function. Neglecting Van-der-Waals effects and focusing on local structure, this method allows rapid manipulation of molecular chains. It was largely used for interactively generating a model sketch of the ribsome (cf. chapter 6).

Similarly, inverse kinematic approaches can be applied to the torsion angle adjustment described in section 2.3.1. Selected atoms are moved interactively and the torsion angles of the connecting molecular chains are recomputed on both sides with inverse kinematic methods, as provided by robotics [HH90, MLS94]. For an implementation as a multi-agent-system see [Boh97].

A molecular dynamics energy minimization is necessary after interactive manipulation with such approximation methods. However, care must be taken with this minimization not to undo what the modeller established interactively. This problem is application specific, and for the ribosome a solution is presented in chapter 6.

[10] `http://www.ks.uiuc.edu/Research/vmd/`

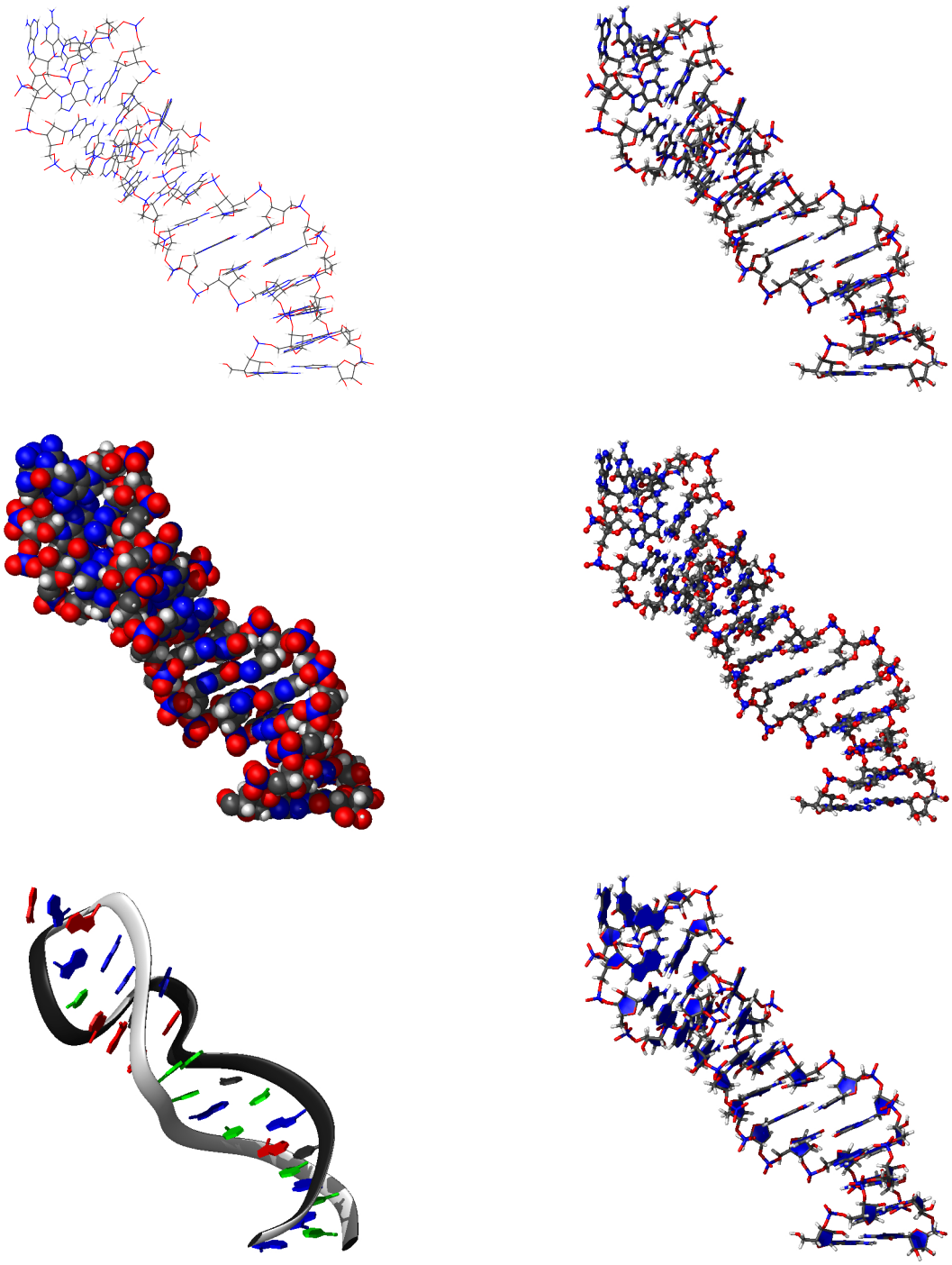


Plate 2.1: The sarcin/ricin loop (cf. section 5.3.3), which consists of ribonucleic acid (section 3.1.2) and is located in the large ribosomal subunit (section 3.2), as an example for visualization techniques:
Top to bottom, left side: wire-representation (bonds are coloured by atom type: carbon is grey, oxygen is red, nytrogen is blue, hydrogen is white), CPK-representation (coloured by atom type) , and ribbon-representation (coloured by nucleotide type); right side: stick-representation, ball-and-stick-representation, and stick-representation with plates added.

Part II

Application to the Ribosome

Chapter 3

The Ribosome: An Overview

In part two the molecular modelling methods developed in the first part of this thesis are applied to the ribosome. The *ribosome* is an intracellular particle which is central to the understanding of molecular genetics. Therefore a brief overview of genetic coding mechanisms, protein synthesis, the role that the ribosome plays in the latter, and an overview of the ribosomal structure are given in this chapter.

The subsequent chapters successively show how this particle has been researched, and how a plausible structural model is constructed on the basis of the gained information.

3.1 Genetics with a Focus on the Ribosome

3.1.1 Proteins and Amino Acids

Proteins are present in all living cells. They serve as the major structural component in animal tissues, are a key part of skin, nails, cartilage, and muscles. Other proteins catalyze reactions, transport oxygen, and serve as hormones. Whatever their function, all proteins are chemically similar, being composed of the same building blocks, called amino acids.

Most amino acids occuring in nature are *2-amino carbon acids*, also called *$\alpha-$amino acids*. A central carbon (C_α) has 4 substituents (figure 3.1): a carboxyl group (COO^-), an amino group (NH_3^+), a hydrogen atom (H) and a side chain (R). This side chain is different for different amino acids (for examples see colour plate 3.1).

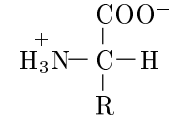
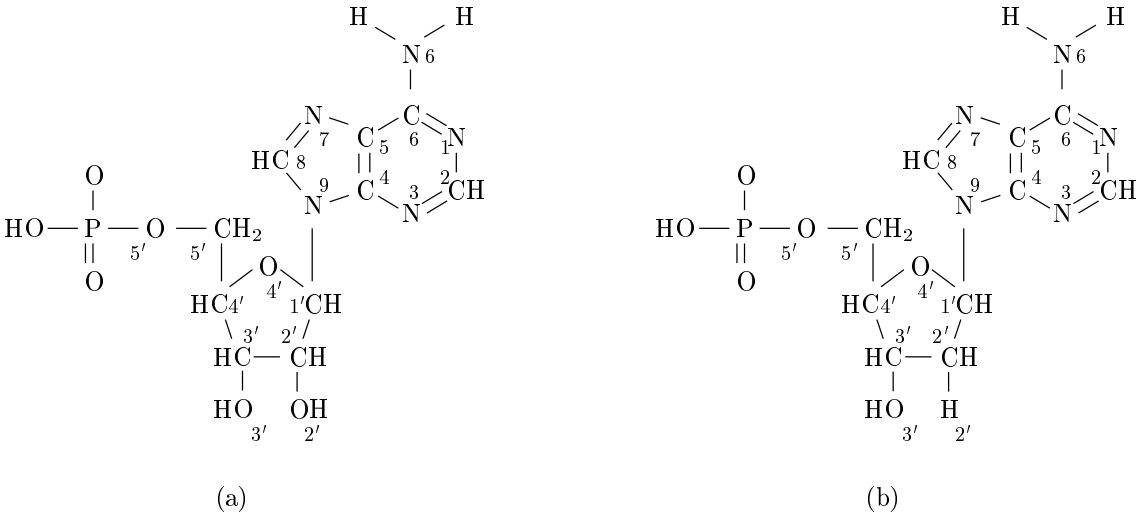


Figure 3.1: α-amino acid

Amino acids can form amide bonds between the α-carboxyl and the α-amino groups. This form of bonding is called *peptide bonding*. It can occur on both sides of the amino acid. Thus, by repetition, strands of amino acids, so-called *peptide chains*, are formed. Proteins are peptide-chains with a specific sequence of amino acids.

In living cells the synthesis of proteins occurs on ribosomes. The process of generation, and the sequence of amino acids is controlled by information carried by nucleic acids.



(a) (b)

Figure 3.2: (a) Adenylic acid RNA nucleotide. The small labels indicate the atomic numbering scheme for nucleotides. The difference between RNA and DNA as depicted in (b) is the oxygen at the 2′ position.

3.1.2 Nucleic Acids

There are two classes of nucleic acids: *desoxyribonucleic acid (DNA)* and *ribonucleic acid (RNA)*. Both types, as molecules, form chains of *nucleotides*. Each nucleotide consists of a base, a sugar, and a phosphate group (figures 3.2, 3.3). The sugar and phosphate groups form the so-called *backbone* of the chain, the bases are protruding at the side. DNA and RNA differ in the way the sugar is built and in the bases. The sugar in RNA is a ribose, in DNA it is deoxyribose. The difference lies at the 2′ position of the sugar ring (figure 3.2): ribose has a hydroxyl (OH) group, whereas deoxyribose has an hydrogen, making RNA chemically less stable than DNA. The bases *Adenine (A)*, *guanine (G)*, and *cytosine (C)* are found in DNA and RNA; *thymine (T)* only in DNA, *uracil (U)* only in RNA. Thymine differs from uracil merely in the presence of a methyl group (cf. figure 3.3). Often strands of nucleic acids are characterized by the names of the bases of succesive nucleotides only, thus ACGU denotes the example strand in figure 3.3.

Base Pairing

Nucleic acids allow *base-pairing*: each base in a strand can be hydrogen-bonded to a complementary base in another strand. The construct is symmetrical, whereby cytosine and guanine are complementary, as are adenine and thymine/uracil. These standard base-pairs are drawn for RNA in colour plate 3.2.

The three-dimensional structure of two such basepaired strands is *helical*, due to regular stacking of the bases. There are several different standard helix forms [Sae84]. An example is given in colour plate 3.3.

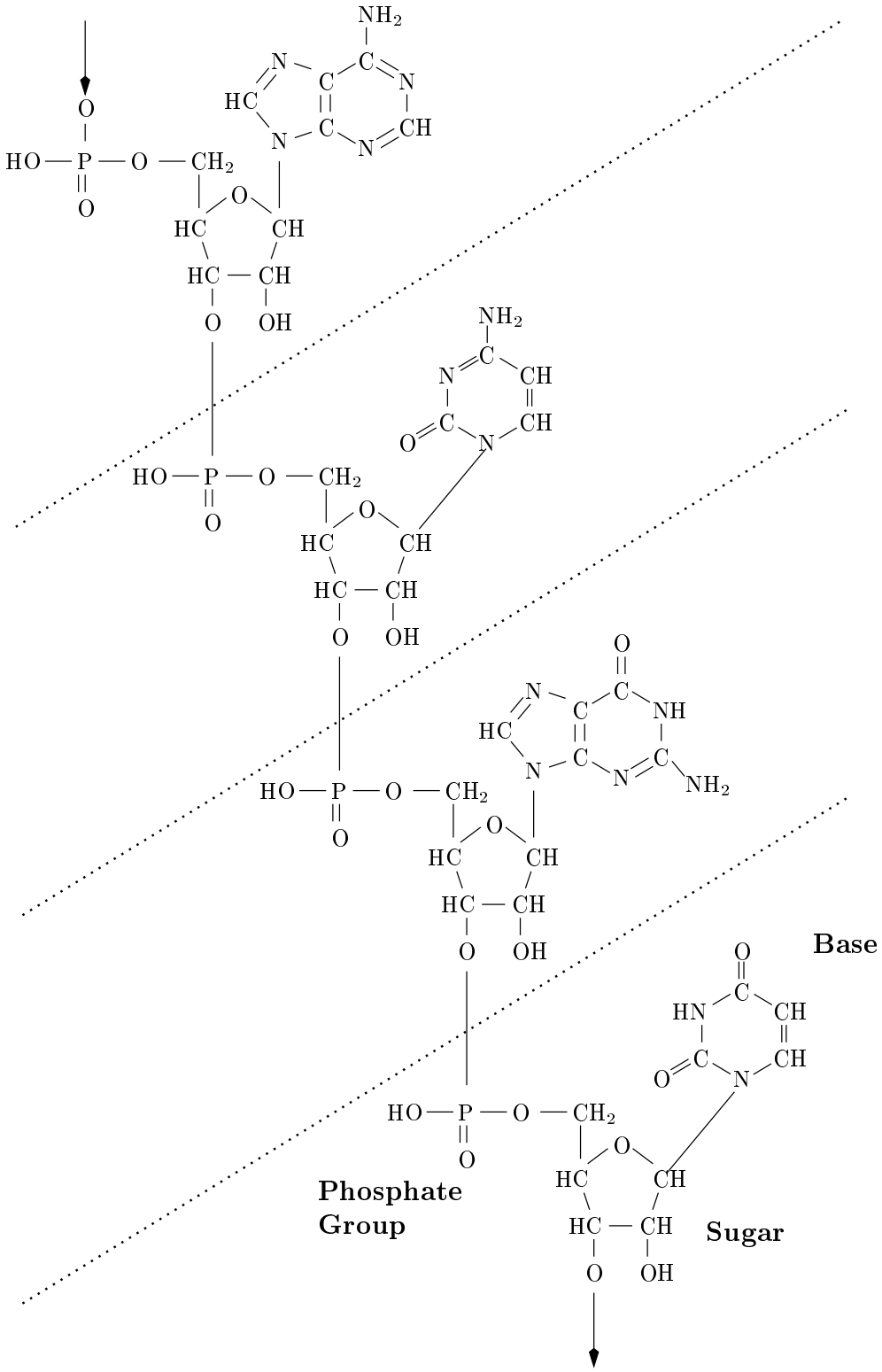


Figure 3.3: Four different nucleotides (conceptually separated by dashed lines) as part of a RNA strand. Each nucleotide consists of a phosphate group, a sugar ring, and a distinctive base (adenine, cytosine, guanine, uracil – top to bottom). The phosphate group and the sugar ring form the backbone connection of the strand (the arrows indicate the prolongation of the strand).

Genetic Code

The base-pairing mechanism allows sequences of nucleic acids to carry information and to code for the amino acid sequences of proteins. The bases of the nucleic acid can be read in triplets, so-called *codons*. For four bases this yields $4^3 = 64$ different code-words, which are sufficient to code for the 20 different types of amino acid occuring in proteins. The difference between 20 and 64 is used to code redundantly and for some control codes.

3.1.3 Molecular Genetics

Genes, Replication, and Transcription

The genetic information of all cells is stored in DNA sequences (RNA occurs as genetic material only in viruses). A piece of DNA that codes for a single function is called a *gene*. Often a gene codes for a protein, i. e. it contains the sequence of amino acids of the protein. Each type of amino acid occuring in the protein is represented as a codon of three sequential bases in the DNA chain.

During cell division the complete information is passed on to the daughter cells, the DNA strands being copied. This is called *replication*. In normal cell functioning to express genes, i.e. to synthesize proteins, the DNA sequences are transcribed into strands of RNA that are often heavily processed and then translated into sequences of amino acids. Both transcription and processing influence what proteins are actually built.

Transcription happens, when an enzyme called *RNA polymerase* binds to a specific base-sequence on DNA. The enzyme unwinds part of the double helix, exposing two single strands of DNA, one of which is then transcribed. As the polymerase moves along the DNA, RNA nucleotides with bases complementary to those of the DNA are added to the growing RNA chain, thus generating an complementary RNA copy of the DNA strand.

In higher organisms this RNA strand has segments containing information coding for protein, called *exons*, and other segments called *introns* that are simply not used. These are spliced out by so-called *small nuclear ribonucleoprotein particles (snRNP)*. The RNA resulting from this process is called *messenger RNA (mRNA)*.

During *translation* the mRNA is read by the ribosome to generate the amino acid sequence of protein. In cells with high protein synthesis activity many ribosomes can read the same piece of mRNA simultaneously, forming a *polysome*.

Translation

The ribosome consists of a large and a small subunit (see subsection on structure). In cells both subunits exist separately and join in the presence of mRNA. The ribosome binds to a *start codon* (AUG) near the $5'$ end of the mRNA and translates toward the $3'$ end by matching each mRNA codon with a helper molecule, specific to that codon. This so-called *transfer RNA (tRNA)* in turn delivers the appropriate amino acid, which is then added to the growing peptide chain. These deliveries are repeated until a mRNA *stop codon* (UAA, UAG, or UGA) is reached. There the

newly synthesized polypeptide is hydrolysed off and the ribosome breaks into its two subunits.

Initiation

Protein biosynthesis is similar in eukaryotic[1] and prokaryotic[2] cells, albeit simpler and better understood in the latter. The following thus refers to the eubacterium *Escherichia coli*.

Initiation is the first phase of translation. Two proteins, called *initiation-factors IF-1* and *IF-3* bind to the small ribosomal subunit. An additional complex of a further factor, *IF-2* and GTP is then bound. This allows the association of the subunit with the mRNA and the binding of a tRNA carrying *N-formyl-methionine (fMet)* to the start-codon. Finally the large ribosomal subunit binds to the complex and the initiation-factors are split off. This fMet-tRNA is at the *P site* of the ribosomal initiation complex.

Elongation and Termination

During the *elongation* phase further amino acids are succesively added to the peptide chain. Elongation can be subdivided into three steps.

I) First the P site of the ribosome is occupied by a tRNA, which carries at it's $3'$-end the complete peptide chain, assembled until then. A second tRNA with an anticodon complementary to the mRNA, exposed at the A site of the ribosome, binds. This second tRNA is loaded with a corresponding amino acid. It binds as a complex with the *elongation-factor Tu (EF-Tu)*, a GTP containing protein. After hydrolysis of the bound GTP to GDP and phosphate, the EF-Tu dissociates. Since the tRNA binding is rather loose at first, the GTP hydrolysis serves as a delay-mechanism to enable checking, whether the correct tRNA is bound. A further protein, the *elongation-factor Ts*, later regenerates the EF-Tu by exchange of the GDP to GTP.

II) Next the peptide bond is synthesized. The *ribosomal peptidyl-transferase* catalyzes the movement of the peptide-chain carried by the tRNA at the P site to the NH_2-group of the amino acid on the tRNA at the A site. The peptidyl-transferase center is located in the large ribosomal subunit.

III) *Elongation-factor G (EF-G)* binds to the ribosome. The hydrolysis of the GTP in that factor delivers the energy to *translocate* the tRNA-peptide complex on the ribosome. It shifts three bases towards the $3'$ end of the mRNA. The two bound tRNAs remain stationary with respect to the mRNA, the tRNA that was in the P site of the ribosome thus being shifted to the E site, where it dissociates. The tRNA holding the peptide chain is concomitantly moved to the P site. Thus, the ribosome is ready for the next elongation cycle.

Termination occurs when a stop codon is reached, as already mentioned above.

[1] Eukaryotes are cells with a nucleus, occurring in fungi, plants, and animals

[2] Prokaryotes are cells without a nucleus, mostly bacteria

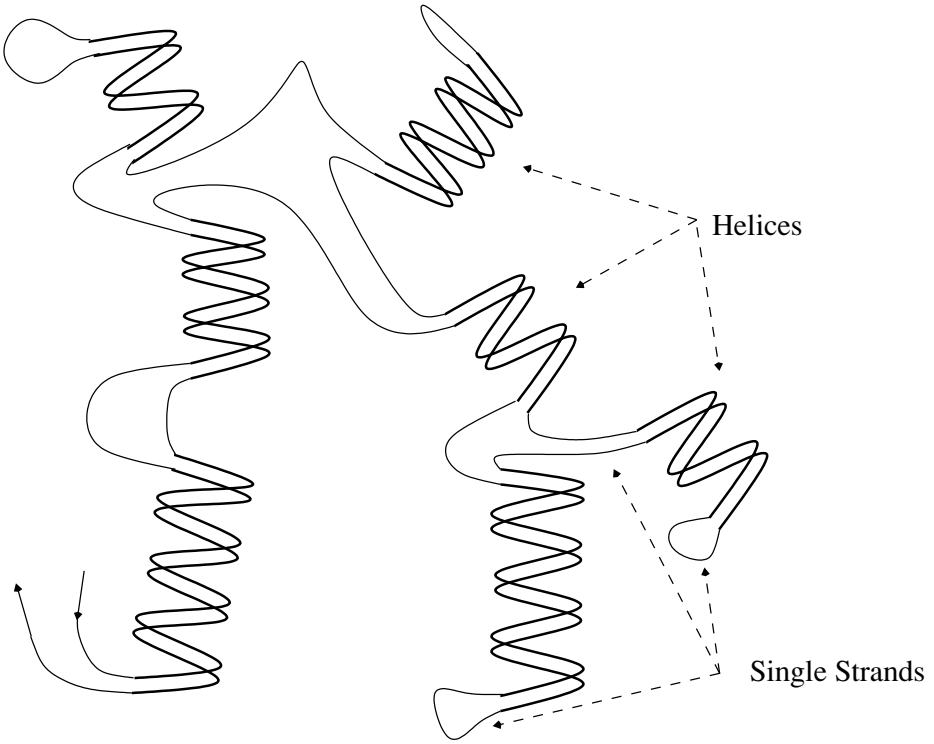


Figure 3.4: Schematical drawing of a ribosomal RNA strand. It consists of helices and connecting single strands.

3.2 Structure of the Ribosome

The ribosome has a large and a small subunit as depicted in colour plate 3.5). Both subunits consist of *ribosomal RNA (rRNA)* and proteins [Bri98]. Instead of specifying the mass of ribosomes and ribosomal components, it is common practice to denote the corresponding sedimentation coefficients in *Svedberg (S)* units.

The focus of our model building studies is on the prokaryotic *E.coli* ribosome, which has a sedimentation coefficient of 70S. Its large subunit has 50S, the small subunit has 30S[3]. The 50S subunit consists of 32 proteins and has two rRNA strands of 23S and 5S. The smaller 30S subunit has 21 proteins and one rRNA sequence of 16S.

For comparison, the eukaryotic 80S ribosome has a large 60S subunit and a small 40S subunit. Its large subunit has three strands of rRNA (5S, 5.8S, and 28S) as well as 49 proteins. The small subunit has one strand of 18S rRNA and 33 proteins.

The ribosomal RNA functions as structural element. Each strand is base paired to form several helices, the segments remaining between the helices are called *single strands* (drawn schematically in figure 3.4).

The following chapters discuss the spatial arrangement of these helices with respect to the single strands and the proteins.

[3]S values are non-additive

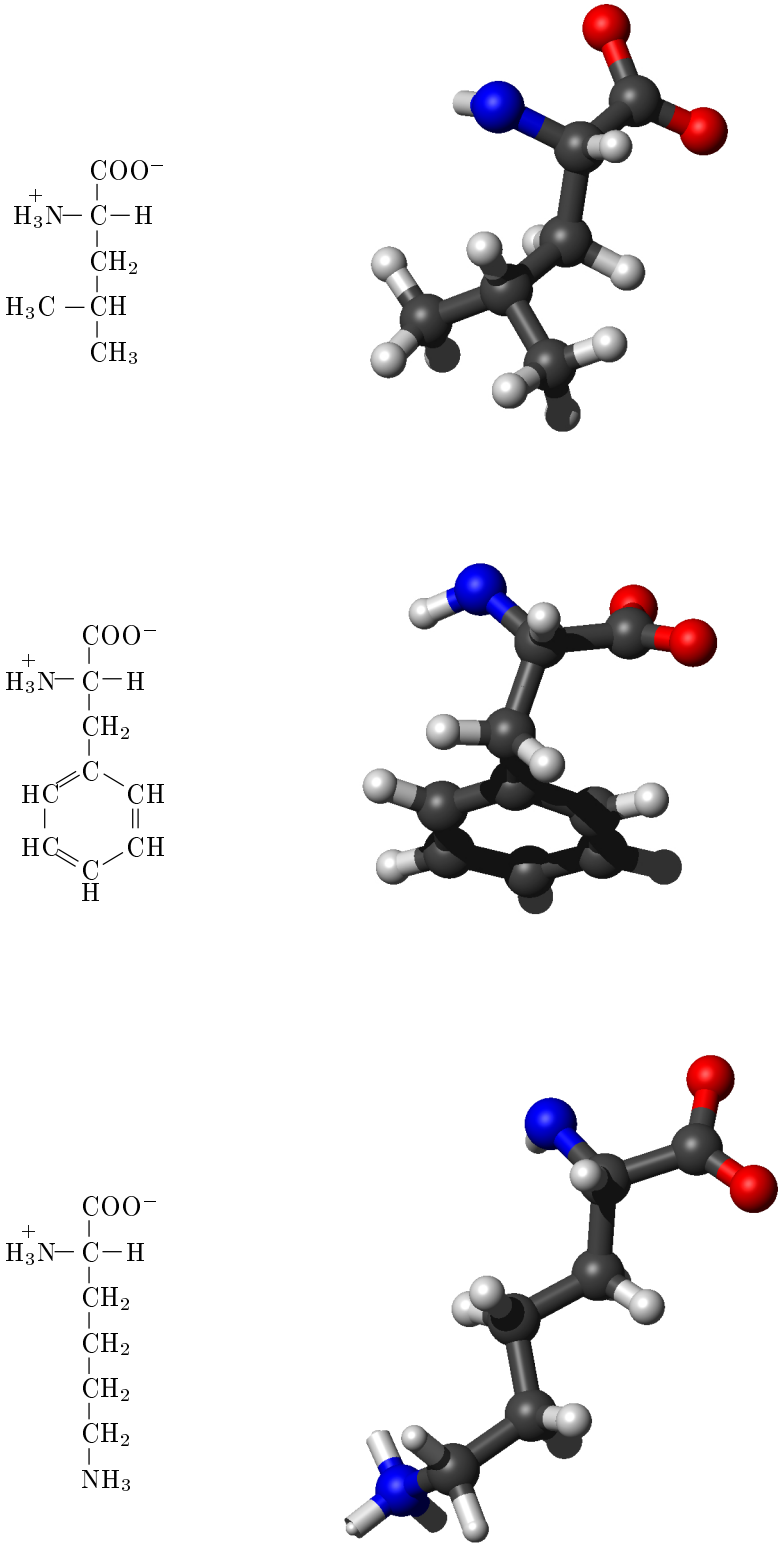


Plate 3.1: Examples of Amino acids. Top to bottom: Leucine (LEU) Phenylanaline (PHE) Lysine (LYS)

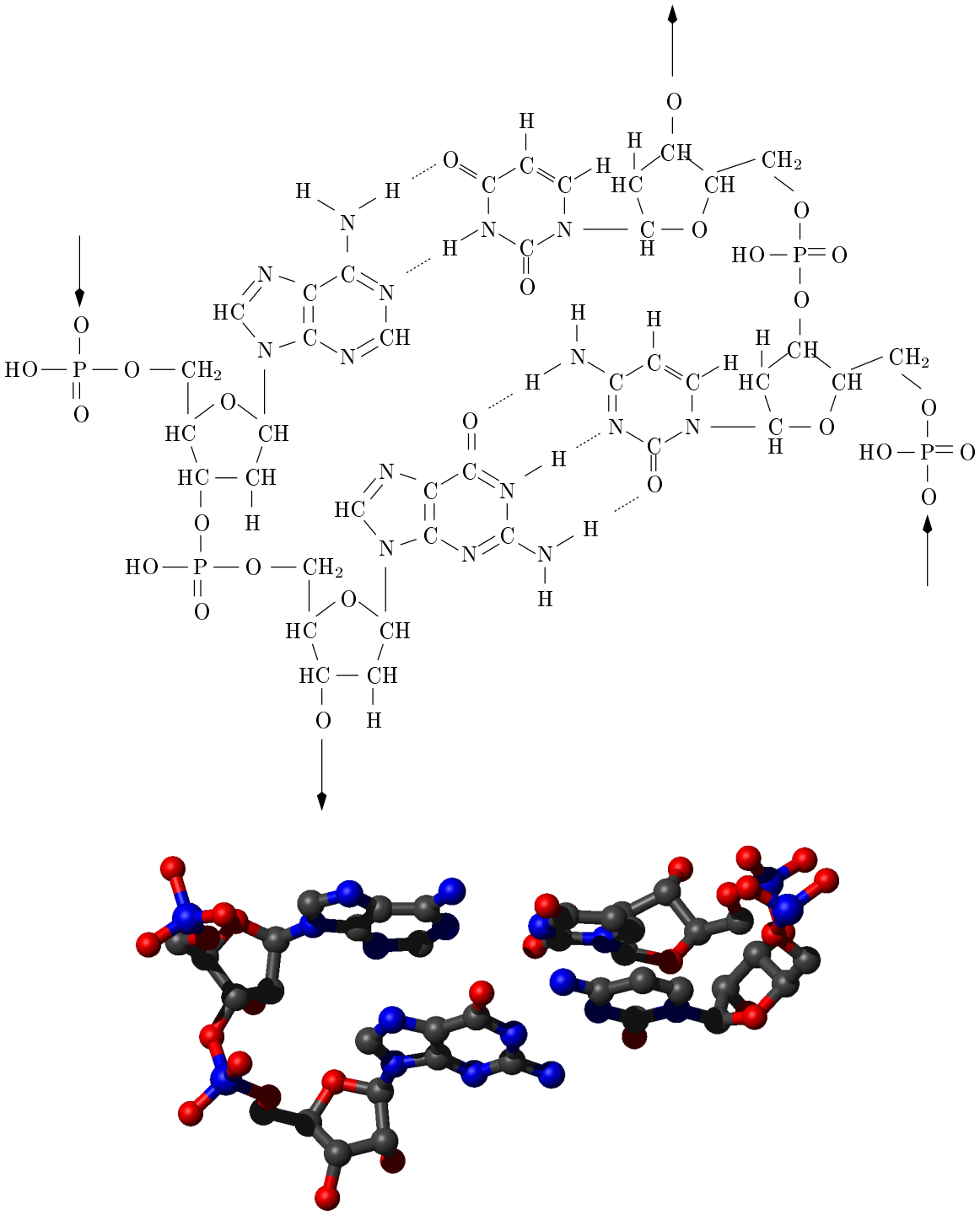


Plate 3.2: Four RNA nucleotides with their bases paired by hydrogen bonding. The upper pair is an A-U, the lower a G-C.

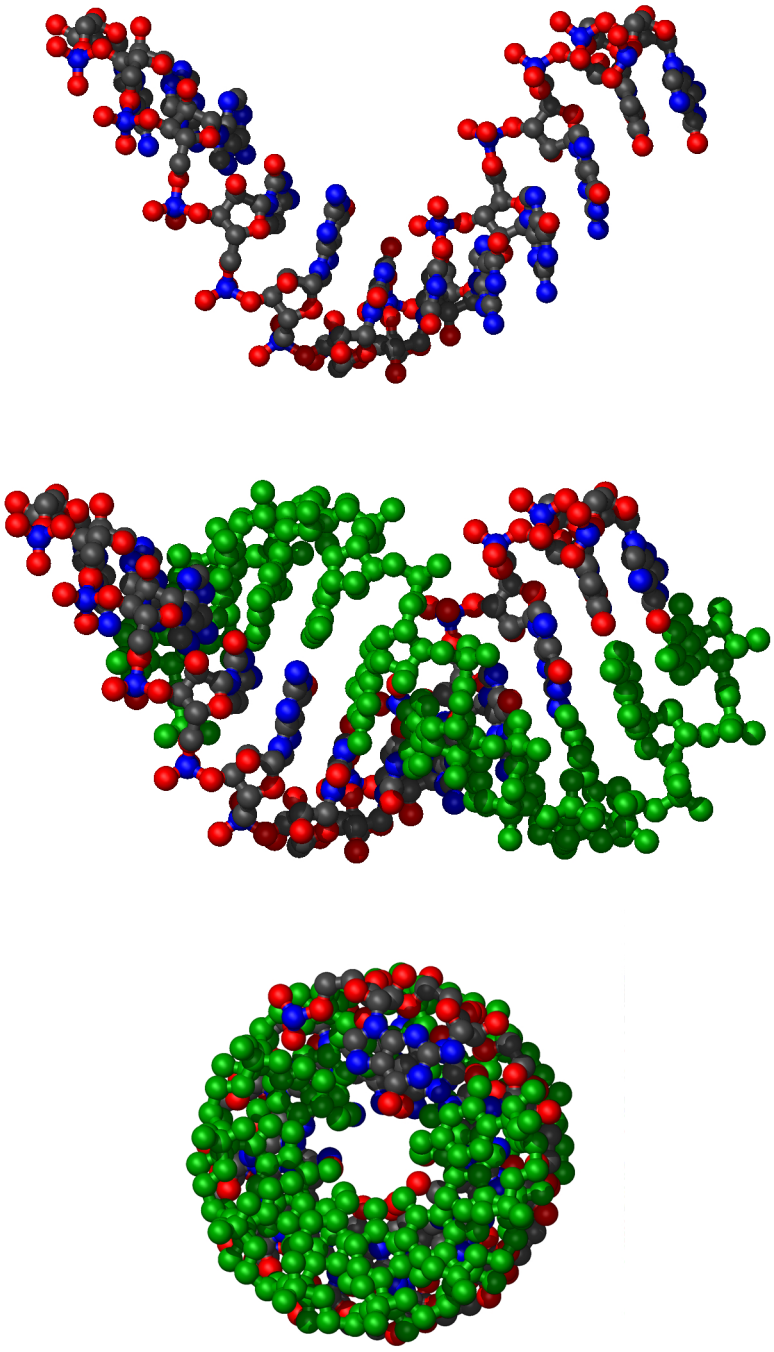


Plate 3.3: Top: a strand of 12 RNA nucleotides. Middle: the same strand base paired with a second strand (green) to form a helix. Bottom: The same helix seen from its axis direction.

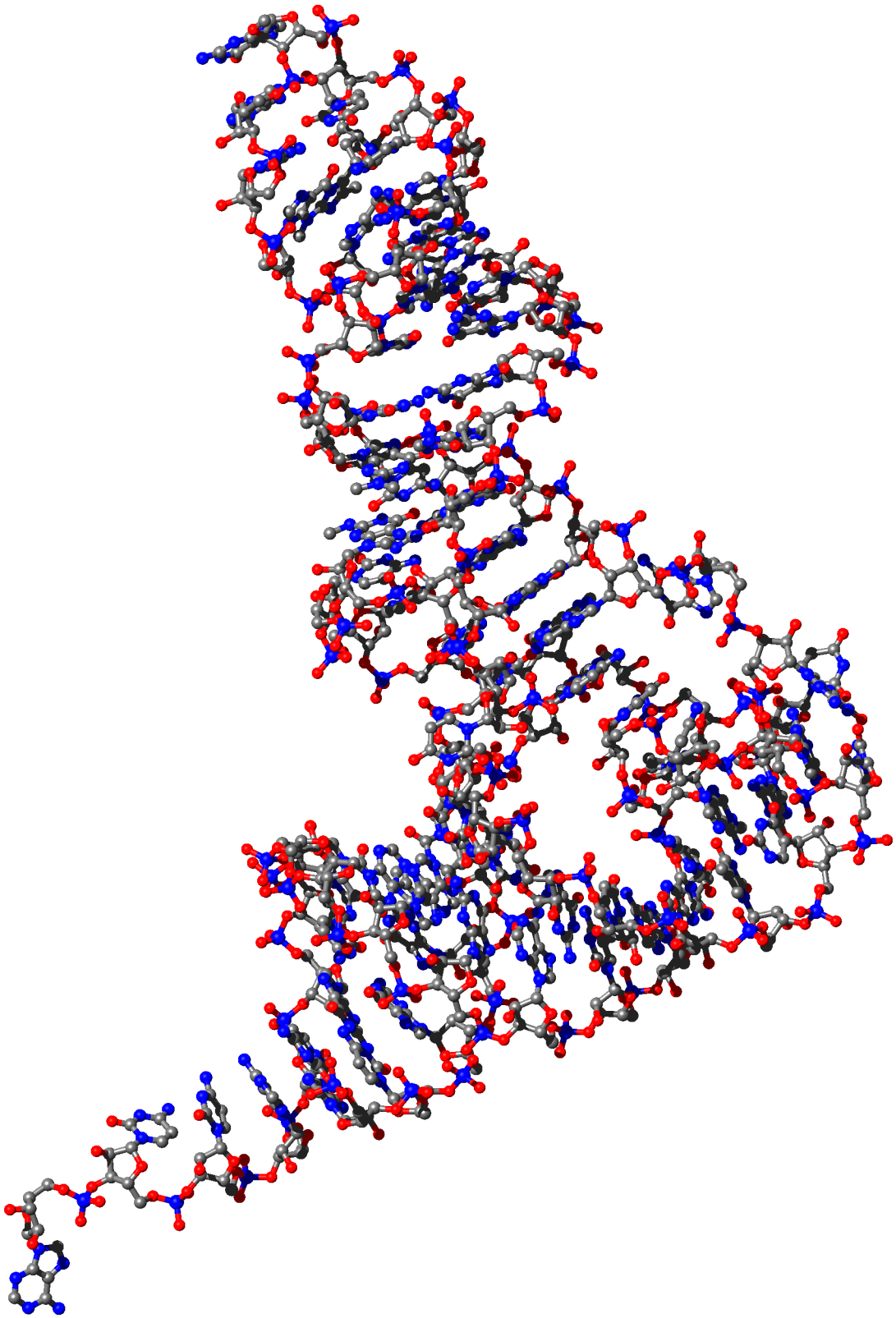


Plate 3.4: Ball-and-Stick model of an *E.coli* tRNA molecule. The anticodon loop is on the upper left corner of the L-shaped structure.

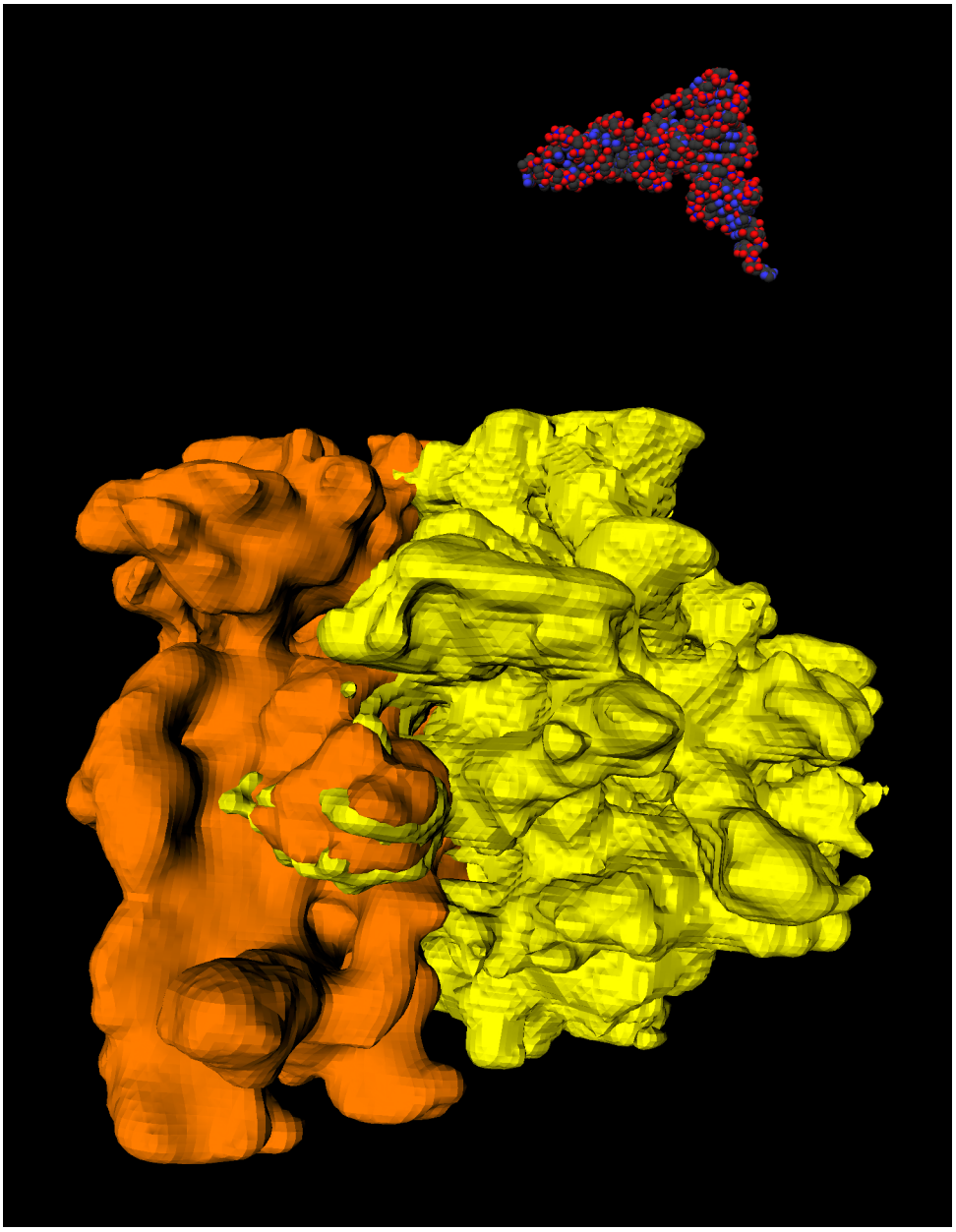


Plate 3.5: The *E.coli* Ribosome with its small subunit(orange) and large subunit (yellow); The contour was prepared by thresholding an electron microscopy density (cf. section 4.3.4). The tRNA molecule from plate 3.4 is shown for size comparison and not located site-specifically

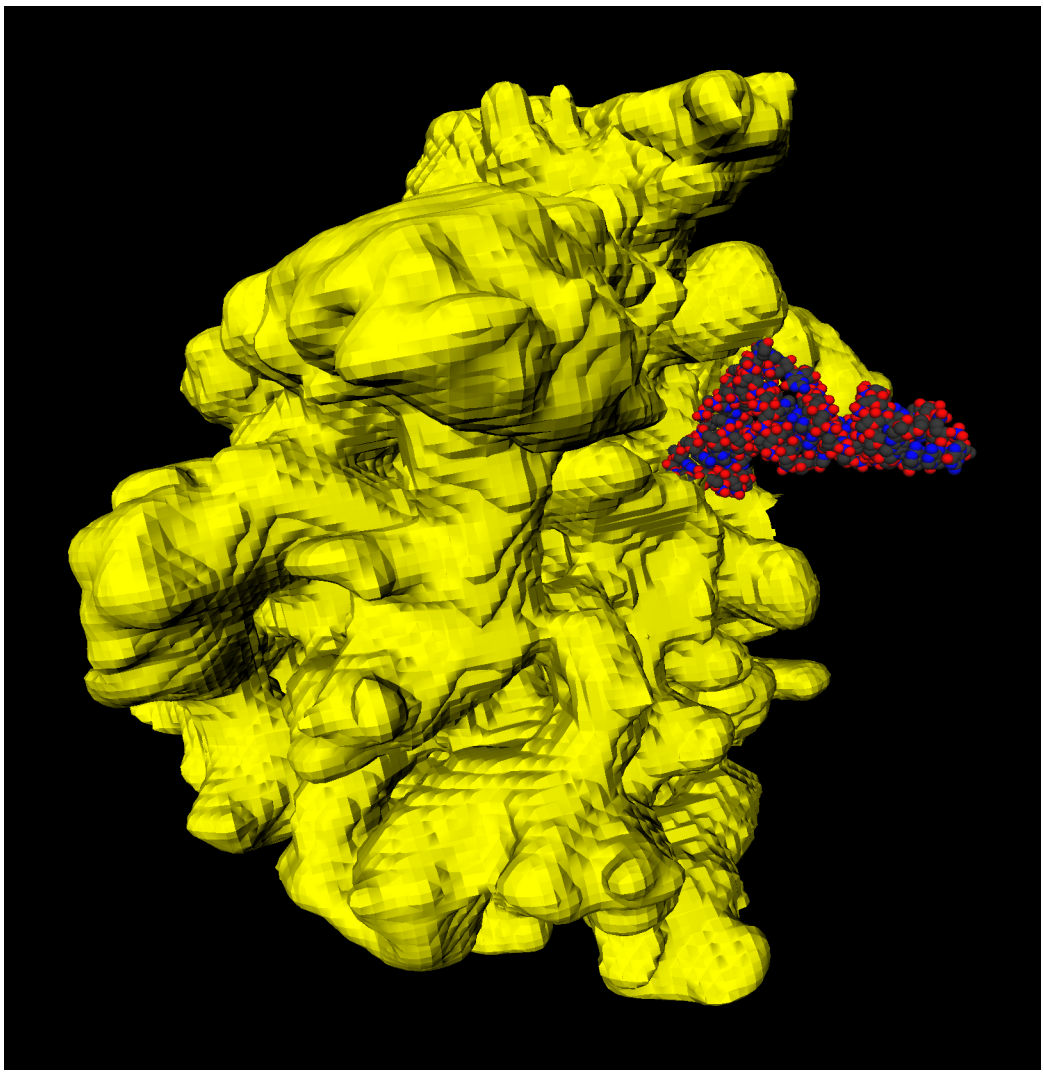


Plate 3.6: Large ribosomal subunit with an A site tRNA; the subunit is rotated 180 degrees around the y-axis with respect to plate 3.5.

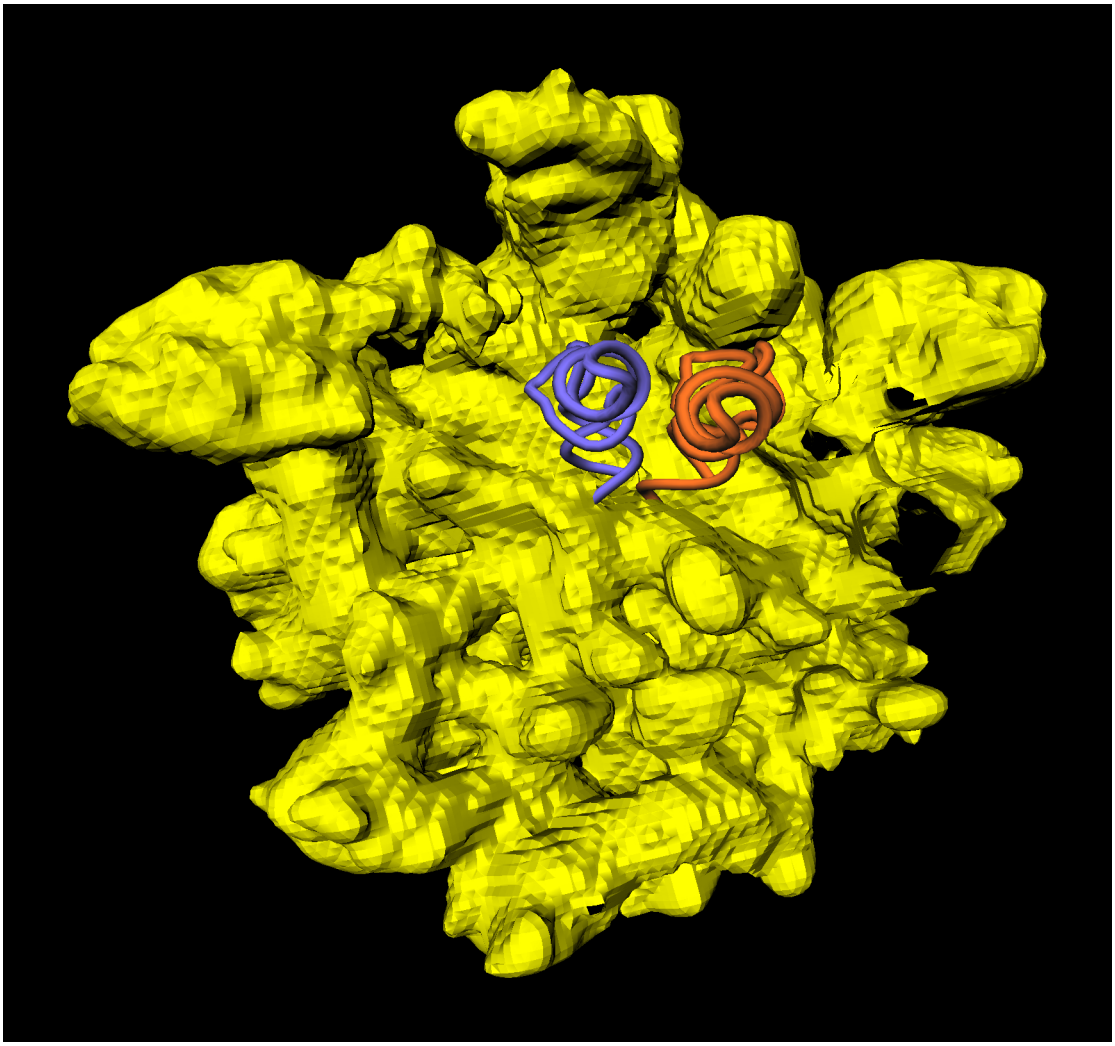


Plate 3.7: Large ribosomal subunit with A site (orange tube) and P-site (purple tube) tRNAs; the subunit is y-rotated 90 degrees with respect to plate 3.5

Chapter 4

The Quest for Structure

A detailed understanding of the function of the ribosome is impossible without knowledge about its structure. Finding the structure is however not an easy task, since even the relatively compact bacterial ribosome has a molecular weight of more than two mega-Daltons. A multitude of techniques thus are being and have been applied to research its structure.

The focus of this chapter is on presenting the techniques used to collect the data that are incorporated into the structural model which is presented in the subsequent chapters.

4.1 Structure Nomenclatures

Molecular biologists refer to primary, secondary, tertiary and quaternary structures. *Primary structure* is given by the sequence of bases within the ribosomal RNA or the sequence of amino acids in a protein. *Secondary structure* is defined by base-pairing interactions within the RNA strands. *Tertiary structure* is defined by the three dimensional folding, whereas *quaternary structure* describes the interaction between individual protein and RNA components.

4.2 Biochemical Methods

To explain where the data on the ribosome originate, some biochemical methods are listed here. The methods are presented briefly to make the following chapters understandable and for more detail the reader is referred to the literature.

By combining these methods, a structural image of the ribosome has been growing historically, and successively parts of the ribosome have been – more and more precisely – localized.

4.2.1 Sequencing

Sequencing is a technique to analyze the primary structure of molecules. As a result the sequences of bases within a strand will be known. Thus in the context of the

ribosome it is mostly of historical interest, since the ribosomal primary structures are known for many species.

A radioactive label is attached to the 5' end of the strand in question. In four parallel setups, one for each base-type, the marked strands are enzymatically or chemically cut into segments. The cutting in each setup is made such that cuts only occur after that specific base-type, with a certain probability for every occurrence. The segments can be sorted by length by gel-electrophoresis. By use of the radioactive label, only those segments that start at the 5' end, and thus were only cut once, can be visualized. Other segments remain invisible. The sorted segments of each type are then displayed next to each other, and the base sequence is read off.

4.2.2 Phylogenetic Comparison

Secondary structure is defined by base-pairing interactions of two segments within the same strand of rRNA. In theory, each base of one strand has a corresponding complementary base in the other, so that only complementary bases can be paired[1].

The "brute-force approach" of searching for complementary sequences of RNA delivers too many possibilities. This is where *phylogenetic comparison* comes in [Bri84]. Here, the basic assumption is that base-pairings within the rRNA are similar across different species. If a base in one strand is exchanged from one species to the other, the complementary base in the other strand must be exchanged as well for the base-pairing to be maintained. It is thus possible to search for such "compensatory" changes in primary structures across species, to find base-pairing interactions.

The method has its drawbacks: Highly conserved structures, i. e. base-pairings with no changes from one species to the other, will not be found. These pairings are often functionally the most important ones. Furthermore regions which are completely different in different species can not be analyzed at all.

In spite of these drawbacks, phylogenetic comparison was one of the major tools to establish the secondary structure of ribosomal RNA.

4.2.3 Foot-Printing

With the *foot-printing* method one analyzes neighbouring regions of RNA and protein quaternary structure, in order to draw conclusions about ribosomal tertiary structure [PN95].

With biochemical methods, it is possible to remove specific ribosomal proteins from the ribosome. Certain chemical agents are known to attack ribosomal RNA at specific places. If a ribosome with removed protein is exposed to such an agent, the agent will attack the ribosomal RNA at additional places. Differences will occur where the protein(s) protected the ribosomal RNA in the complete ribosome.

4.2.4 Immune Electron Microscopy

Immune electron microscopy was used to determine the location of individual proteins and of some characteristic regions of ribosomal RNA strands within their sub-

[1] This only holds in most cases, but not necessarily for all helices.

units [Wit83, Wit85, SSM86]. Antibodies can be raised against certain modified nucleosides of the rRNA. The positions of these antibodies on the ribosome have been identified, and thus the concerned regions of the rRNA were localized within the ribosome by electron microscopy.

The localized regions were used as fixed points in attempts to elucidate the spatial arrangement of the ribosomal RNAs and proteins within their subunits.

For example such experiments have been done with the nucleoside N^6, N^6-dimethyladenosine, which occurs at positions 24 and 25 from the 3' end of the 16S RNA. Similarly, the 7-methyl-guanosine at position 526 of the 16S RNA was mapped and the 3' ends of 16S RNA (184–186), of 23S RNA (28, 79), and of 5S RNA (79, 187) as well as the 5' end of the 16S RNA were localized on the *E.coli* ribosome.

4.2.5 Cross-linking

Cross-linking is a method to establish neighbourhood relations within the ribosomal tertiary or quaternary structure [Wit83, SAZB86, SJB95]. For example, RNA can be linked with a bifunctional chemical reagent to other regions of RNA in situ. After the link is established, the RNA is extracted and partially digested with nucleases. The fragments are separated by two-dimensional electrophoresis and analyzed for cross-links. After identification of which segment of primary structure was linked to which other segment, this yields information about spatial closeness: One base of the first fragment must have reacted with one function of the reagent (which has a measurable maximal length), and the other function of the reagent has reacted with a base in the other segment. Similarly RNA segments can be tested for closeness to protein, and protein–protein relations have been established. Of course, different reagents are different in length. In 1997 a database of the published ribosomal crosslinks (DRC) was created, and is accessible at `http://www.mpimg-berlin-dahlem.mpg.de/~ag_ribo/ag_brimacombe/drc/` .

4.2.6 Artificial Mutation

Mutation, i. e. the changing of genetic information, can be used in several ways to analyze function and structure of biomolecules. For example a particular base in the RNA of the small ribosomal subunit could be changed from adenine to guanine. By growing the cells with the altered genes one sees whether the change is lethal, reduces growth, or does not affect cell development. It can thus be concluded whether the change is important for the cell functionality.

Another recently developed strategy is to introduce specific mutations, so that certain helices in the ribosomal RNA are elongated. Since helices are visible with 3D electron microscopy (cf. section 4.3.4), this clearly identifies where certain parts of the secondary structure are localized spatially [P. Sergiev, R. Matadeen, R. Brimacombe, M. van Heel, in preparation].

4.3 Biophysical Methods

4.3.1 Neutron Scattering

Under certain conditions, deuterium deflects coherent neutron beams in a different manner from normal hydrogen. This effect, referred to as *neutron scattering*, can be used to localize deuterium-containing groups within molecules [Mue93].

Ribosomes grown in the presence of deuterated water contain only deuterium instead of hydrogen. Biochemically these ribosomes can be taken apart and recombined with some normal ribosomal components. If only two proteins in the newly assembled ribosome remain deuterated, the distance between their mass centres can be estimated with the neutron scattering method. The technique was used by Moore and coworkers to elaborate a distance map of the proteins within the small ribosomal subunit [CKEM88].

4.3.2 X-ray Crystallography

X-ray crystallography is a method to determine high resolution structures of molecules: Crystals are generated, illuminated with X-rays and the diffraction patterns are analyzed [Sae84, Joh85].

In a crystal molecules are regularly arranged in a grid. The molecule is arranged within a *unit cell* by application of space group[2] symmetry operations. Translations of the unit-cell in all three dimensions define the crystalline array.

For analysis, the crystal is illuminated with parallel, monochromatic X-rays. By movement of the crystal a great number of lattice planes within the crystal are brought into reflecting position in order to measure the corresponding diffraction patterns. The symmetry and systematic extinctions of the diffraction patterns furnish information about the space group symmetry within the unit cell, and the distances between reflections give unit cell dimensions.

The many spots in a diffraction pattern are individually numbered with Miller indices hkl, where h, k, and l are integers. Each spot is generated by diffraction from a set of parallel crystal lattice planes, which are in a reflecting position towards that spot. The set of planes is also indexed hkl. The corresponding glancing angle ϕ_{hkl} depends on the vertical distance d_{hkl} of these planes, according to *Bragg's law*: $2 * d_{hkl} * sin\phi_{hkl} = n\lambda$, where n is the order of reflection hkl, and λ is the wavelength (e.g. 1.542 Å for the commonly used CuK_α radiation)[3].

All the beams hkl diffracted from the crystal are characterized by an amplitude $|F_{hkl}|$ and a phase α_{hkl}. By densitometric scanning of the photographic film or by means of a scintillation counter mounted on an automatic diffractometer, the amplitude can be measured, while the phase information is lost (*phase problem*) and has to be estimated.

[2] *Space groups* are symmetry groups in three dimensional space, with small translations excluded as symmetry operations. A *symmetry group* is the group in the mathematical sense of all symmetry operations of a body.

[3] Obviously, the better the resolution to be obtained, the smaller the d_{hkl}, and the wider ϕ_{hkl}. In crystals of macromolecules the reflections tend to fade away toward the edge of a diffraction pattern, limiting the resolution in general.

From the amplitudes $|F_{hkl}|$ and estimated phases α_{hkl}, corresponding to the parallel planes, the density information ρ_{xyz} within the unit cell can be computed with an inverse Fourier Transform: $\rho_{xyz} = 1/V \sum_h \sum_k \sum_l |F_{hkl}| e^{i\alpha_{hkl}} e^{-2\pi i(hx+ky+lz)}$. V is the volume of the unit cell, which is divided into Grid points xyz for computational purposes.

Crystal generation techniques are nontrivial and resolution is a function of crystal quality and crystal size. For small molecules with a molecular weight of less than 2000 Daltons the resolution is usually better than 1 Å, leading to clear separations of individual atoms. For larger molecules the resolution is not as good, rendering the data subject to interpretation with help of model building studies.

Currently two types of crystallographic approaches are being applied to the ribosome: Complete crystallization and crystallization of individual components.

Several groups have succeeded in generating crystals of complete ribosomes [BFN+98, YF98]. Because of the size of the ribosome, the resolution currently is around 9 Å, enforcing the need for interpretation.

A number of ribosomal proteins as well as RNA protein complexes have been succesfully crystallized and analyzed (cf. section 5.3.3). With particles only a fraction of the size of the complete ribosome, much higher resolution is possible. This yields highly precise pictures of parts of the ribosome. Whether the structure of such complexes within the ribosome is completely identical to their structure crystallized separately is a question that remains to be answered.

4.3.3 Nuclear Magnetic Resonance Spectroscopy

Another technique for determining high resolution structures is *nuclear magnetic resonance (NMR)*[4] spectroscopy: The parameters *chemical shift* θ, *spin-spin coupling* J, *spin-lattice relaxation time* T_1, and *nuclear Overhauser enhancement (NOE)* can be interpreted in terms of intra-molecular conformational characteristics and inter-molecular interactions [Sae84, RJ85]. Information about hydrogen bonding and base stacking is deduced from chemical shift data θ. Details on molecular conformation can be calculated from the spin-coupling constants J, which are related to torsion angles within the structure. Spin relaxation times T_1 and the nuclear Overhauser effect can be used to determine the orientation of a base relative to its sugar within nucleic acid structures.

All details of an NMR analysis can be computationally combined to reveal a picture of the structural characteristics of a molecule. The result is a set of possible 3D arrangements, often overlayed in the same picture. An example is given in colour plate 4.1.

The complete ribosome is currently too big to be analyzed with NMR spectroscopy methods and they are being applied to functional parts, as with the crystal structures just mentioned (cf. section 5.3.3).

[4]Nuclear magnetic resonance is the selective absorbtion of electromagnetic radiation by an atomic nucleus in the presence of a strong static magnetic field

4.3.4 3D Electron Microscopy

In *electron microscopy* electron rays are transmitted through a specimen, and projection images are recorded on *micrographs*. With projections from different angles, information about a particle's three-dimensional structure can be computed. Different projections can be obtained either by multiple exposures while tilting the specimen within the microscope or by exploiting random orientations of many particles within the specimen.

Radiation damage gradually destroys biological particles during the image generation process. One is forced to collect a large number of images under "low-dose" conditions, hence exploiting the random orientations of particles in the specimen is favourable.

The 3D information is computed in a three step process: I) signal enhancement and classification of views, II) angular reconstitution, and III) 3D back projection. These steps are iterated, because the results from back projection help to improve the signal.

Signal Enhancement

In a set of noisy images of a particle, the noise at any position varies from image to image, but the desired feature information is the same. By averaging images, the signal is enhanced with respect to the background noise, and the *signal-to-noise ratio* (*SNR*) is improved.

Micrographs contain projection images of many orientations of the particle, but only images of the same orientation can be averaged. It is thus necessary to sort the images before averaging. This can be done by *multivariate statistical classification*, using *correspondence analysis* to find the main features of the images [vH84].

For correspondence analysis (analogous to *principal component analysis*, or *Karhunen-Loève transform*), images are represented as points in a hyper-dimensional space, with as many dimensions p as there are pixels in one image. The intensity value $x_{i,j}$ of each pixel j of an image i is used as the coordinate along the j axis of the hyper-space. Given n images, a new orthogonal basis $\{b_1, \ldots, b_p\}$ of the hyper-space is then sought, such that the first basis vector b_1 points into the direction of highest inter-image variance. The next basis vectors b_k are constructed perpendicular to the others $b_l, l < k$, in the direction of maximum remaining inter-image variance[5]. Since the basis-vectors correspond to a decreasing amount of inter-image variance, the first few vectors (typically 2 to 8) fully characterize the most important differences in the image set. The remaining axes – the total number still equals the number of pixels – describe less and less of the inter-image variance and may therefore be disregarded. By transforming the original images coordinates into this new basis of hyper-space, the most important features of each image can be now be represented by only a few numbers.

An automatic classification method, such as *hierarchical ascendent classification (HAC)*, is then used to find sets of images which can be averaged into *characteristic views* with an enhanced SNR.

[5] the basis $\{b_1, \ldots, b_p\}$ can be calculated with an eigenvector eigenvalue analysis, for details see [vH84].

Angular Reconstitution

The characteristic views represent projections of the particle in different directions. The angular reconstitution technique allows these so far unknown directions to be computed.

Two different projections of a 3D object onto planes (2D projections) always have a projection onto a line (1D projection) in common [vH87, vHOH$^+$ss]. This line, the "common tilt axis", is perpendicular to the projection directions of both projections. Given three different projections planes p_1, p_2, p_3 with common tilt axes C_{12}, C_{13}, C_{23} it is possible to analytically determine the angles between the planes and thus their relative orientations (figure 4.1). Since direct solving of the analytical equation turns out to be unstable, a brute-force search over all possible particle orientations is used instead [vHOH$^+$ss].

The projection directions relative to particle orientation, specified by Euler angles (appendix A.5), can thus be computed.

3D Back Projection

With *back projection* three-dimensional information can be computed from the projection images. This operation can be thought of as uniformly smearing back, or projecting back, the density from each pixel of the projection image along the projection direction, into the volume to be reconstructed. The summation of these back projections represents an approximation to the original object. However, each point in the approximation will be blurred to a star shape, and a correction function must be applied [HvH86, SOD$^+$ss].

Specimen Preparation

As described in section 3.1.3 the ribosome is a "machine" with a multitude of states, and different states correspond to different structures. It is therefore necessary to biochemically fix the ribosomes in one of their functional states, which can be accomplished in several ways. For example, the reconstruction for the model discussed in chapters 5 and 6 makes use of a "ternary" complex of elongation-factor Tu, aminoacyl-tRNA and GTP. Stable Ribosome complexes can be formed on a near natural messenger RNA by first binding fMet-tRNA to the P site in the presence of initiation-factors, then adding the antibiotic kirromycin, and finally the EF-Tu-GTP-Phe-tRNA complex. The antibiotic stalls the multi-step process of A site binding immediately after codon recognition and GTP hydrolysis [SRRA$^+$97].

These samples are flash frozen in solution, so that the individual particles are randomly oriented in the resulting matrix of vitreous ice.

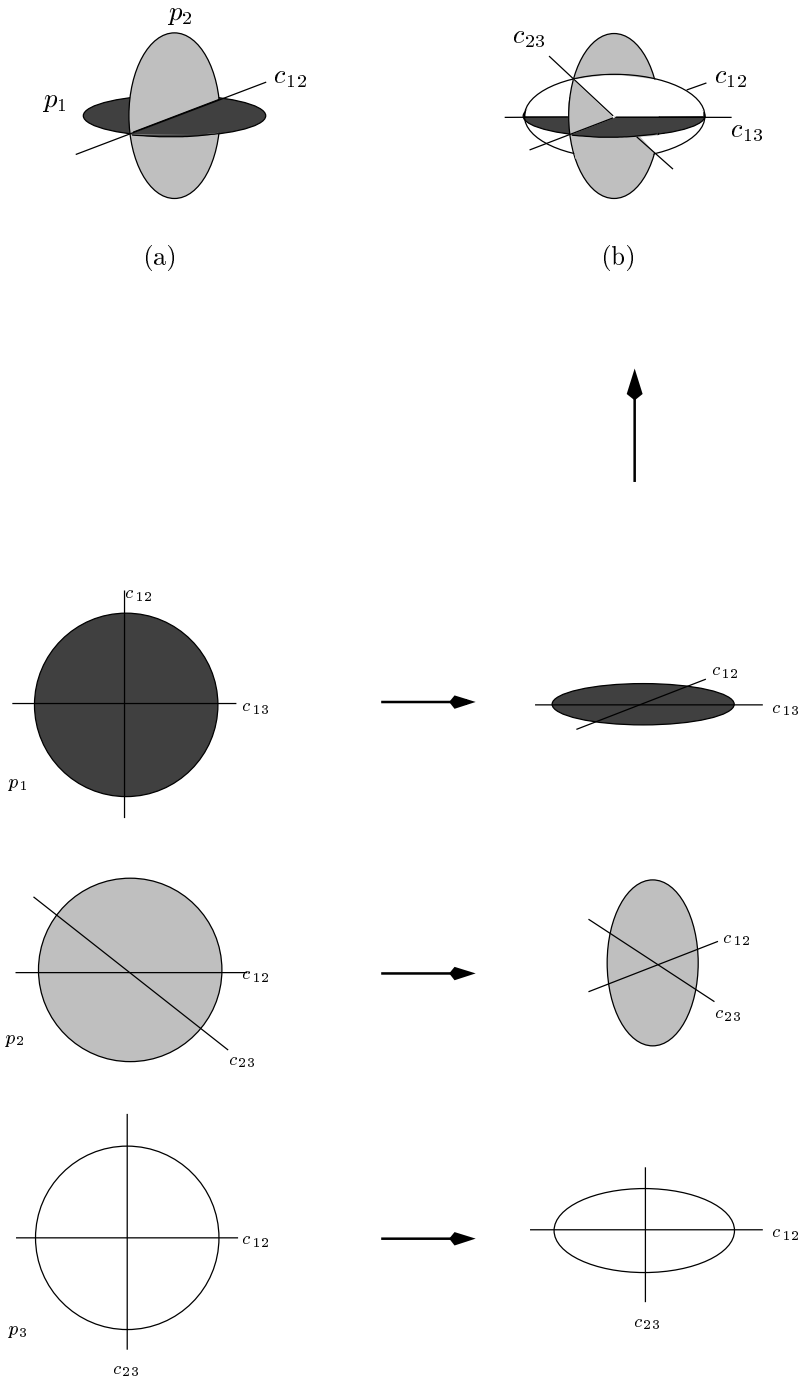


Figure 4.1: Angular Reconstitution: Two 2D Projections of a 3D object always have a 1D projection in common (a). Therefore three 2D projections (c) are sufficient to find the respective spatial orientations and thus the projection directions (b).

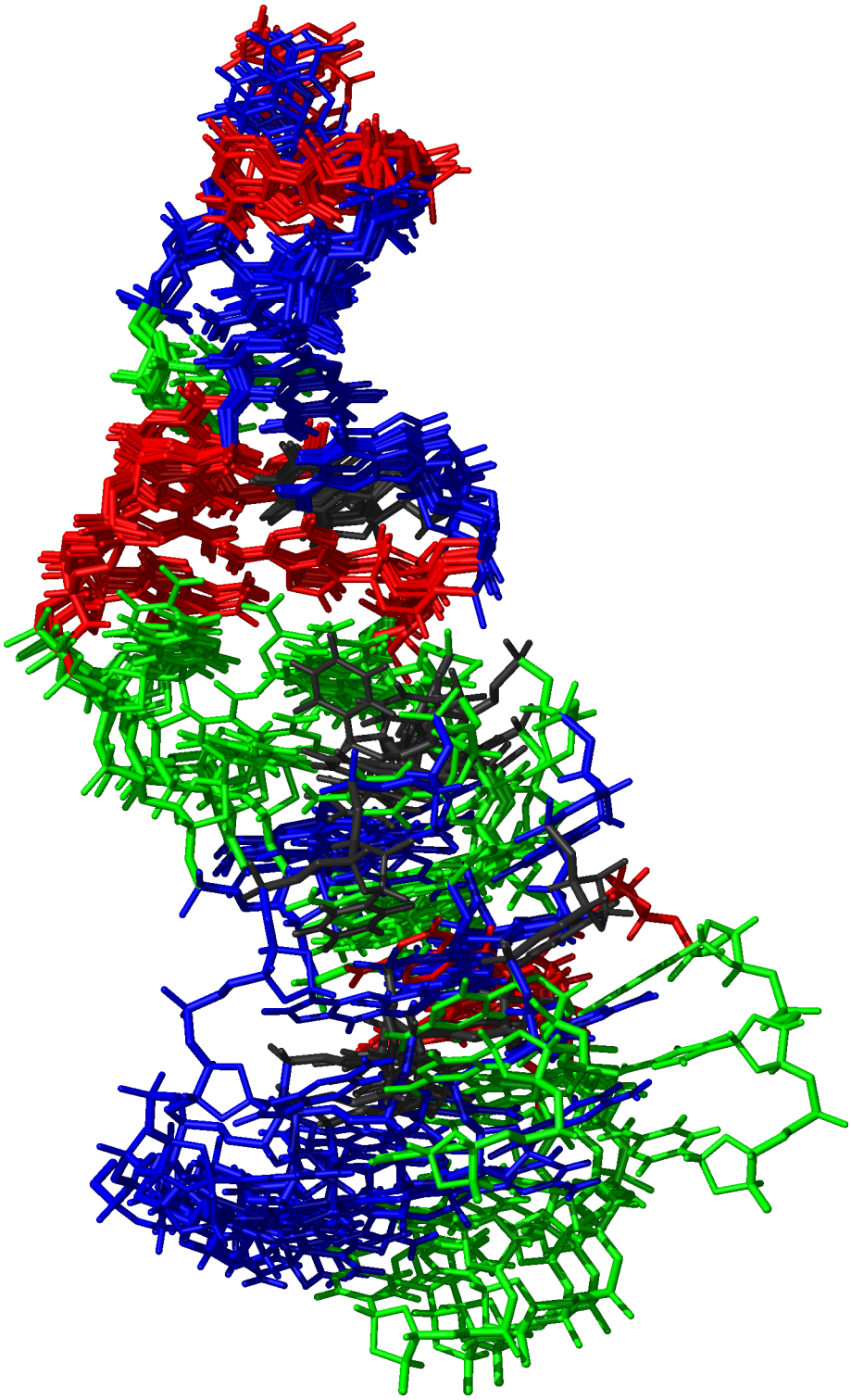


Plate 4.1: Superposition of six slightly different structures that satisfy the NMR data for the sarcin/ricin loop in the large ribosomal subunit [SM95]. The nucleotides are coloured by the types of their bases: adenine (red), cytosine (green), guanine (blue), and uracil (grey).

Chapter 5

Structural Properties of the Ribosome

As pointed out in chapter 3, the ribosome consists of RNA and protein. The structural properties of RNA and of protein have been thorougly researched by methods explained in chapter 4.

For the computational investigation of the ribosome which is carried out in chapter 6, by the methods developed in the first part of the thesis, a preparational step is necessary, which is described in this chapter. The energy potential that was defined in general in section 1.2 needs to be adapted to the structural properties of the ribosome.

On a detailed level, it has to be specified which atoms are bound, what the desired bond lengths should be, which bond angles are important and which are the desired values of these angles; furthermore what the force constants are, and which Van-der-Waals effects matter.

On a higher order level, the groups of atoms that can be clustered into rigid objects for the clustered molecular dynamics technique (section 1.5) are specified.

Additionally, schemes are developed which permit a particle of the complexity of the ribosome to be specified.

5.1 Ribosomal RNA

The ribosomal RNA consists of several long strands, the primary sequence of which is kown by sequencing methods, briefly described in section 4.2.1. The base-pairing of the strands is known by phylogenetic and other approaches. This results in the *secondary structure maps* depicted in figures 5.1 to 5.3. The low-level properties of the particle can thus be specified nucleotide by nucleotide, following the strands.

Low-Level Information per Nucleotide

The sequence of different nucleotides within a strand is known. For each type of nucleotide a pattern is specified, defining which atoms are covalently bond to which other atoms within the nucleotide, and also which atoms within the nucleotide must be connected to atoms in the previous and in the next nucleotide of the strand.

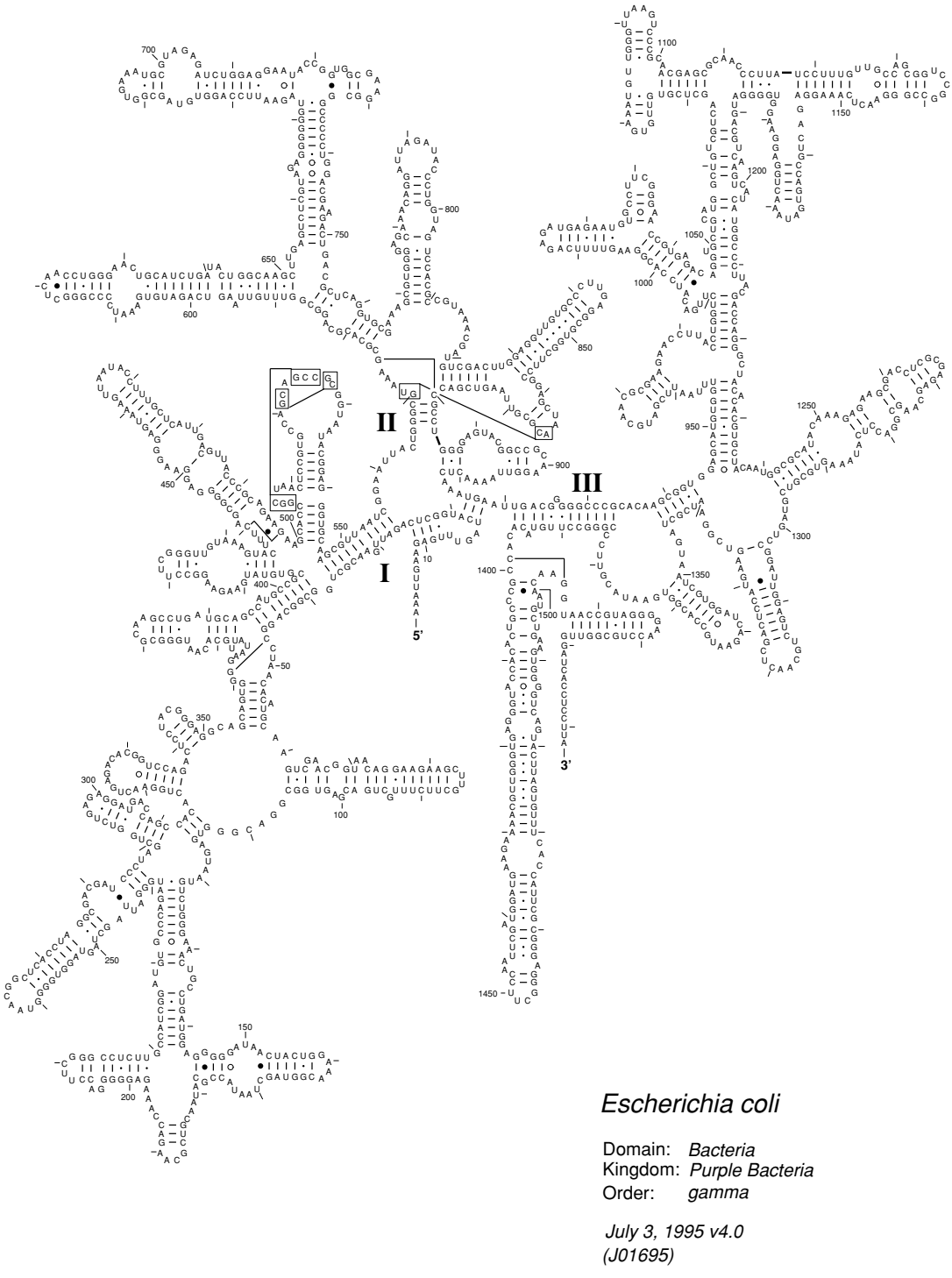


Figure 5.1: Secondary structure of the small ribosomal subunit. Computerized version [Gut94].

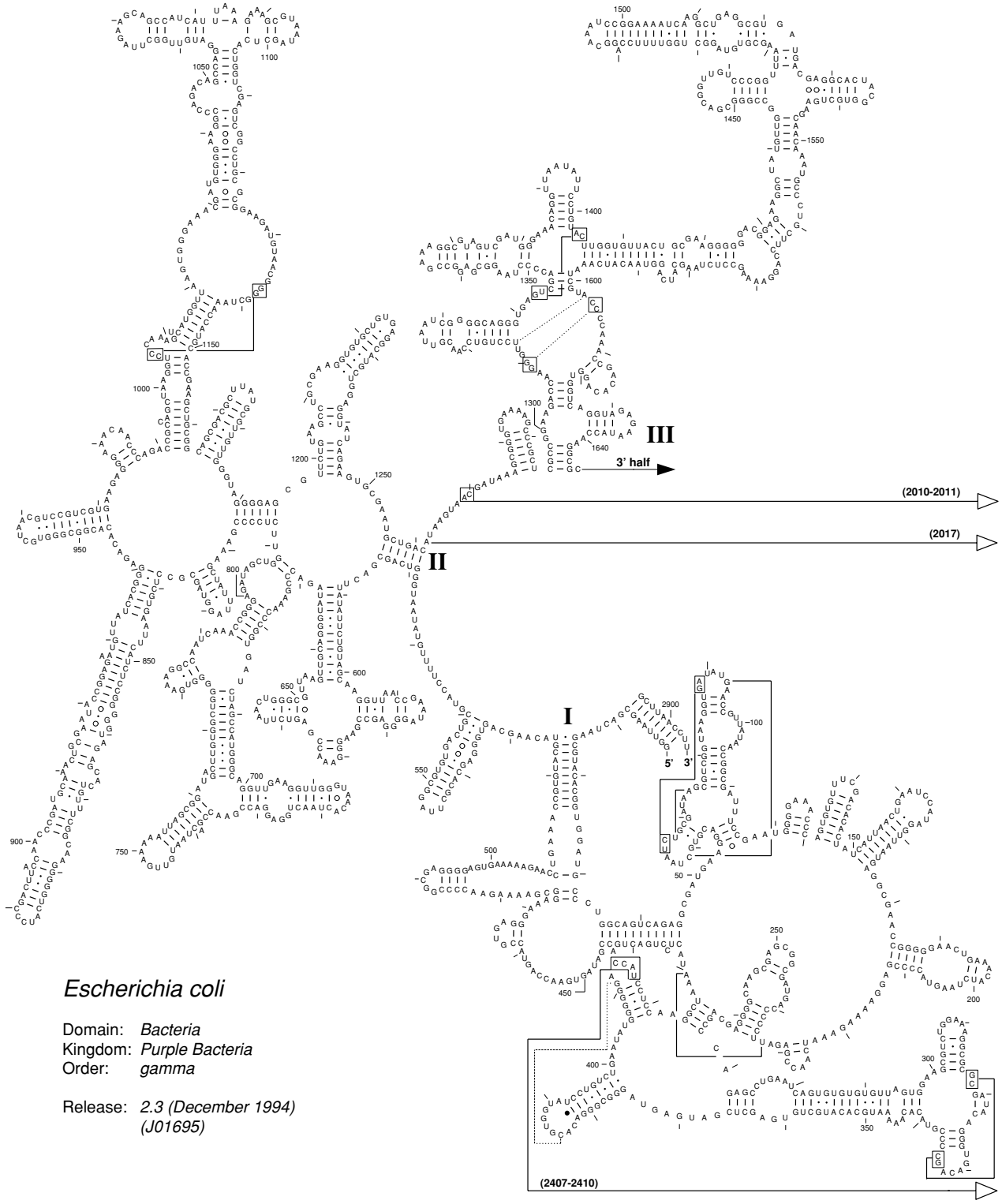


Figure 5.2: Secondary structure of large ribosomal subunit, 5′ half [GGS93].

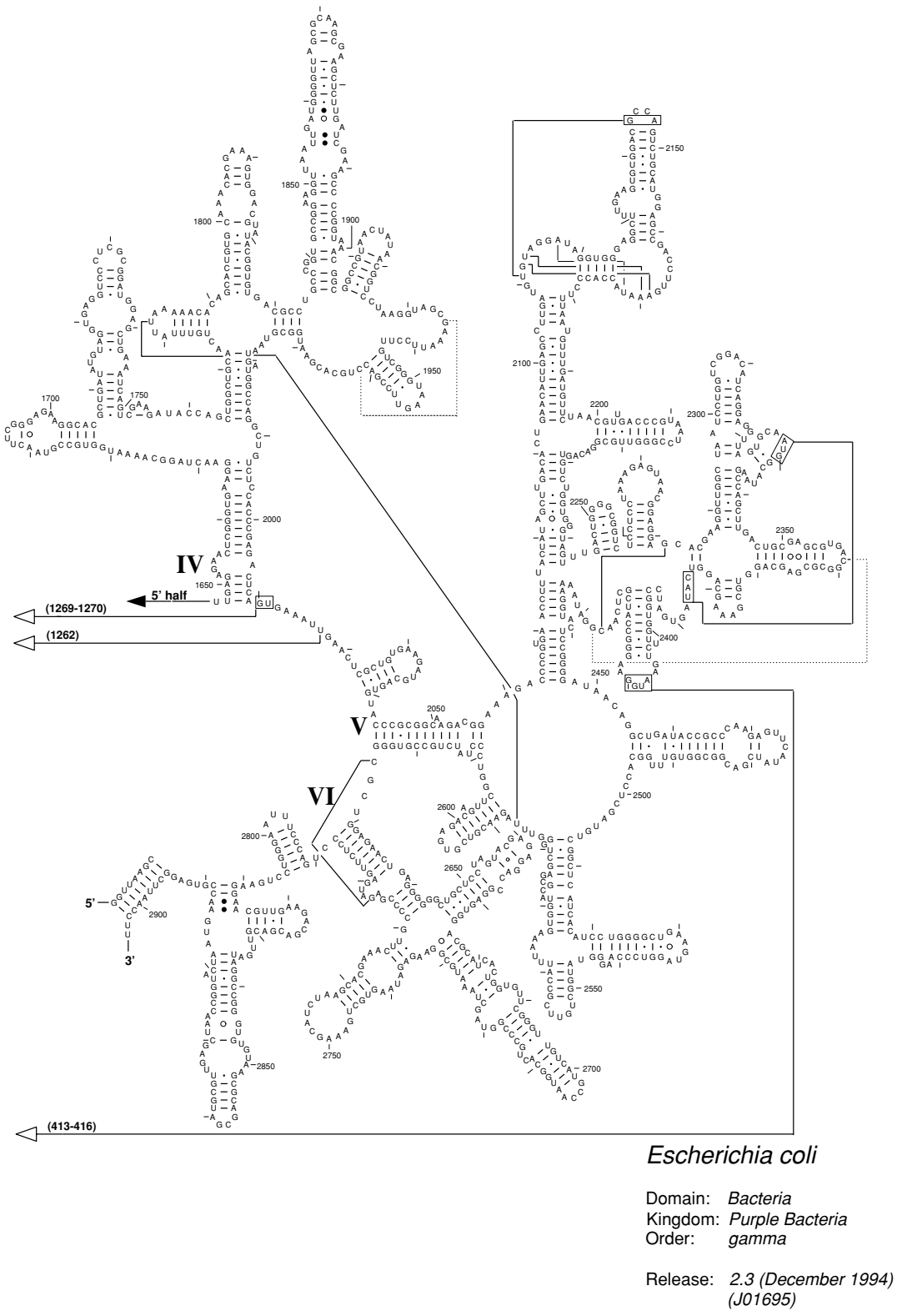


Figure 5.3: Secondary structure of large ribosomal subunit, 3′ half [GGS93].

Type of Atom a_i	Type of Atom a_j	A_{ij} $\left[\frac{\text{kcal}\cdot\text{Å}^{12}}{\text{mole}}\right]$	B_{ij} $\left[\frac{\text{kcal}\cdot\text{Å}^{6}}{\text{mole}}\right]$
H	H	$4.46 * 10^3$	46.7
H	C	$38.0 * 10^3$	128.0
H	N	$27.0 * 10^3$	125.0
H	O	$25.1 * 10^3$	124.0
H	P	$154.0 * 10^3$	346.0
C	C	$286.0 * 10^3$	370.0
C	N	$216.0 * 10^3$	366.0
C	O	$205.0 * 10^3$	367.0
C	P	$1090.0 * 10^3$	1000.0
N	N	$161.0 * 10^3$	363.0
N	O	$153.0 * 10^3$	365.0
N	P	$835.0 * 10^3$	990.0
O	O	$145.0 * 10^3$	367.0
O	P	$796.0 * 10^3$	995.0
P	P	$4080.0 * 10^3$	2710.0

Table 5.1: Coefficients A_{ij} and B_{ij} of the Lennard-Jones potential [Sae84]

Along with it the desired bond lengths and force constants for the bond springs are enumerated as well as the bond angles and the force constants for them. This is schematically drawn for an adenylic acid nucleotide in figure 5.4, the precise description and values for the standard nucleotides being listed in appendix B.

5.2 Van-der-Waals Interactions

In sections 1.2 and 1.3 Van-der-Waals interactions between atoms were defined. Which atoms are involved in this energy and force calculation was left open there and is specified here for the ribosome, as follows.

Van-der-Waals energies occur between all atoms a_i, a_j that are not bound covalently. The force constants A_{ij} and B_{ij}, that specify the attractive and repulsive behaviour in equation (1.6) on page 8 depend only on the type of the atoms a_i and a_j involved. Values valid for RNA molecules are listed in table 5.1. In practice, values at distances more than 9 Ångstrom are neglected. Accordingly, the radii that control the switching function sw, defined in (1.7) are chosen to be $r_{\text{on}} = 8\text{Å}$, and $r_{\text{off}} = 9\text{Å}$.

For atoms that are clustered into the same rigid object, Van-der-Waals forces, like all other forces, cancel out. Therefore they are not considered, although for modelling feedback purposes, the corresponding energies can be evaluated.

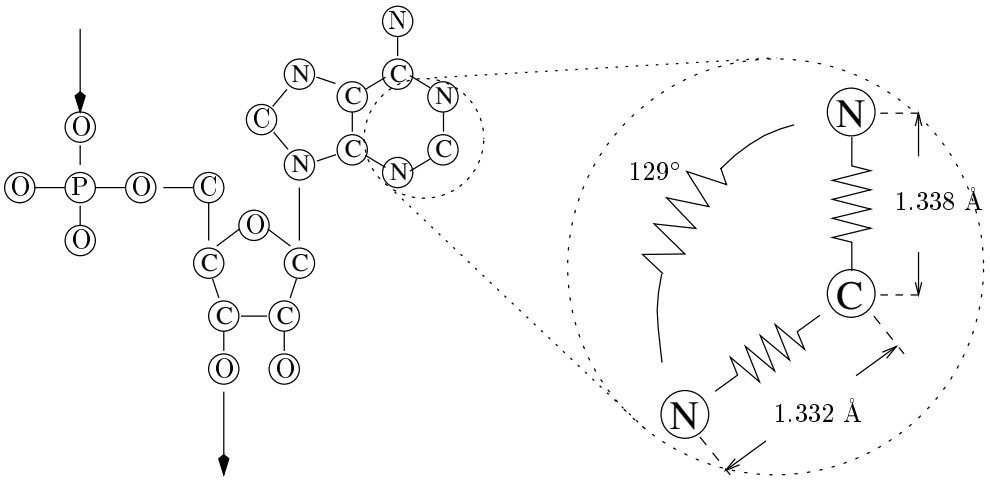


Figure 5.4: For each type of nucleotide a pattern is defined, specifying which atoms are covalently bound, and what the desired bond lengths, bond angles, and force-constants are. The figure shows all bonds defined for an adenylic acid nucleotide. The magnification shows exemplary bond lengths and angle values.

5.3 Clustering

The minimization methods presented in the first part of this thesis allow clustering of atoms into rigid objects. Clustering is possible at different levels of resolution – even simultaneously within one particle.

Which groups of atoms are clustered strongly influences the model, and thus control of the clustering becomes part of the modelling process and will be discussed in chapter 6. Here groups of atoms are suggested that can potentially be clustered into rigid objects during the modelling process.

For RNA in general, each strand can be treated at several levels of resolution: All-atom, sub-nucleotide, nucleotide, and supra-nucleotide. The all-atom level of resolution does not need further discussion as there are no rigid objects. The other levels of resolution are discussed in the following sections.

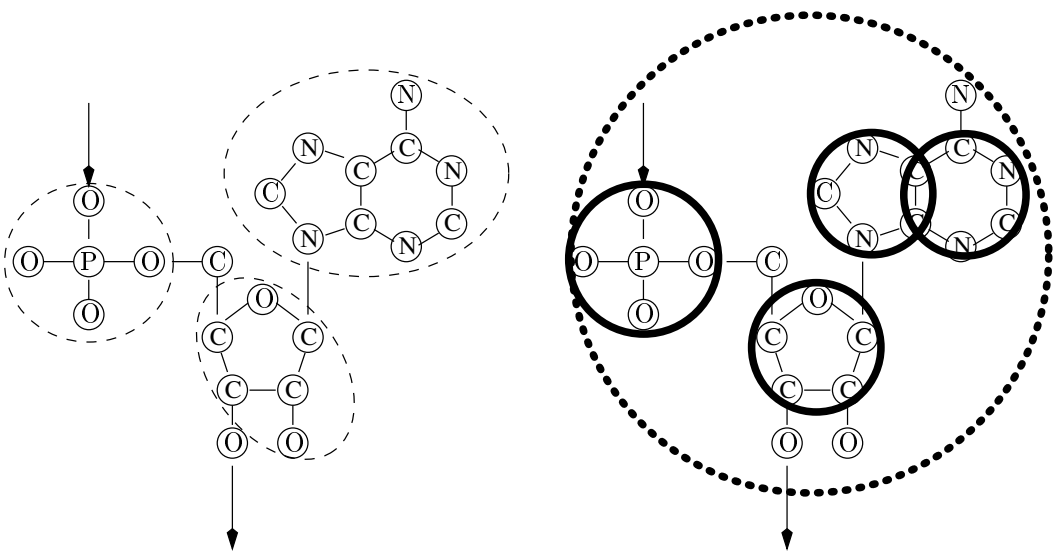
5.3.1 RNA Nucleotides

Groups

At the sub-nucleotide level groups are specified that can be potentially clustered into rigid objects. As shown in figure 5.5 (a), these are the phosphate group, the sugar and the base of each nucleotide. At the nucleotide level of resolution the complete nucleotide becomes a rigid object.

Pseudo-Atoms

Pseudo-atoms are then generated that can be facultatively attached to the rigid objects defined by the groups. The pseudo-atoms have a somewhat larger radius, comprising several other atoms. In an early stage of the energy minimization process a few pseudo-atoms can be used to replace many regular atoms of the grouped object



(a) (b)

Figure 5.5: a) Each nucleotide is subdivided into groups (drawn as dashed ellipses), that can be potentially clustered into rigid objects. The atoms that are not within groups form single atom objects. b) Additional pseudo-atoms can be attached to the rigid objects. X-type pseudo-atoms (black circles) function to keep certain spaces free. Y-type pseudo-atoms (dotted circle) are used to simplify the computational process at an early stage.

and thus reduce computational complexity. At a later stage pseudo-atoms are used to prevent other atoms from occupying certain space. The latter pseudo-atoms, defined at the sub-nucleotide level, are called X-type, and in a standard nucleotide there is one in the phosphate group, one in the sugar ring, and one or two in the base. The former pseudo-atoms are called Y-type and there is one comprising the complete residue. For adenylic acid, this is depicted in figure 5.5 (b), and the complete specification is listed in appendix B.

5.3.2 RNA Helices

Helices are defined by base-pairing interactions between opposite strands. Helices are rather stable in structure, and therefore top candidates to be clustered into rigid objects. Each helix inherits the properties of the residues it contains. Furthermore it can be clustered into a rigid object and new types of pseudo-atoms are attached.

Clustering

To define a helix as a rigid object, all the nucleotides that are involved in the base-pairing interactions of the helix are clustered. Structures of this kind range from 2 to 26 nucleotides in size in the generated model of the ribosome.

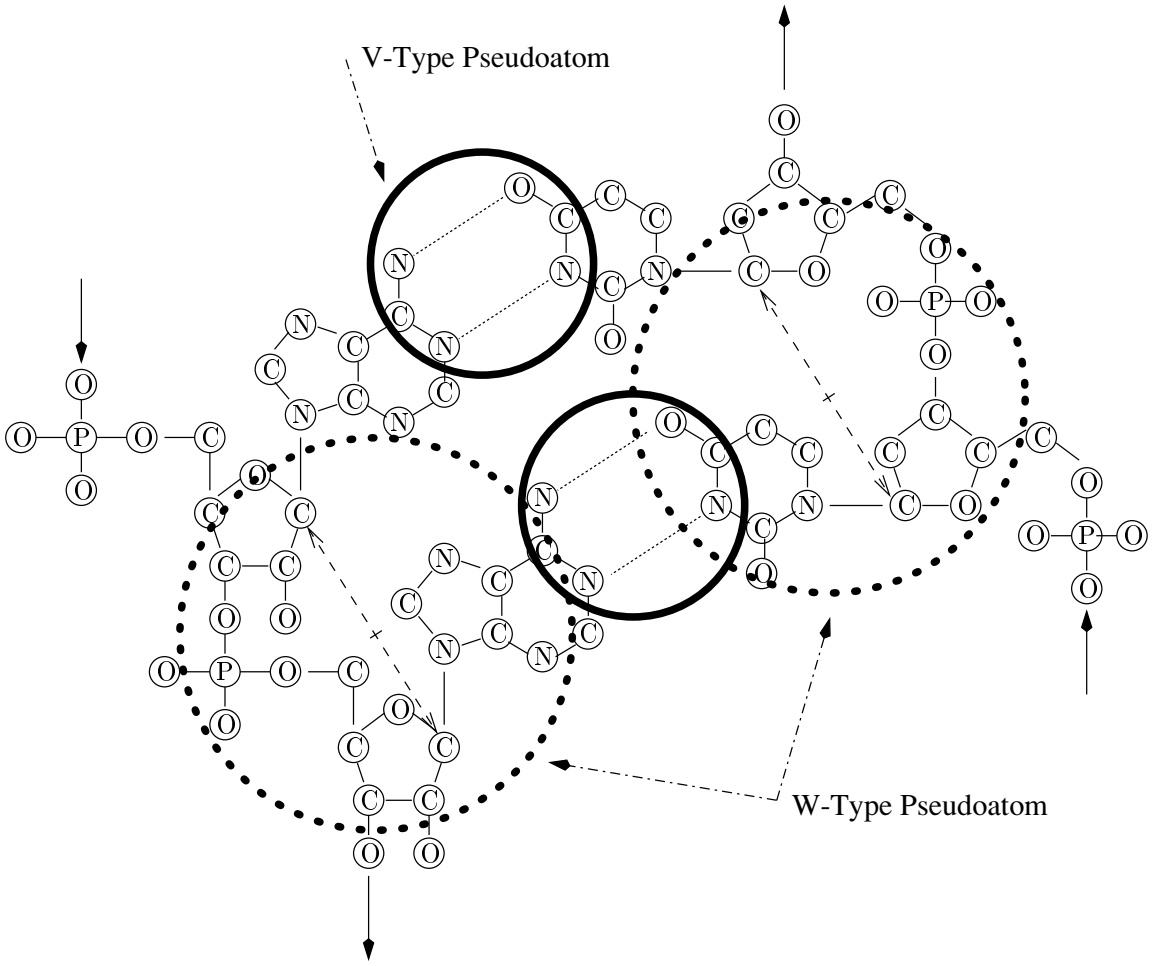


Figure 5.6: Two new types of pseudo-atoms are generated for RNA helices: V-type pseudo-atoms are placed between the hydrogen bonds, and W-type pseudo-atoms are placed in the middle C1′ atoms of succesively stacked residues.

Protein	Species	Method
S4	*Bacillus stearothermophilus*	X-ray, NMR
S5	*Bacillus stearothermophilus*	X-ray
S6	*Thermus thermophilus*	X-ray
S7	*Bacillus stearothermophilus* and *Thermus thermophilus*	X-ray
S8	*Bacillus stearothermophilus* and *Thermus thermophilus*	X-ray
S15	*Bacillus stearothermophilus* and *Thermus thermophilus*	X-ray, NMR
S17	*Bacillus stearothermophilus*	NMR
L1	*Thermus thermophilus*	X-ray
L6	*Bacillus stearothermophilus*	X-ray
L7/L12	*Escherichia coli*	X-ray
L9	*Bacillus stearothermophilus*	X-ray, NMR
L11	*Bacillus stearothermophilus*	NMR
L14	*Bacillus stearothermophilus*	X-ray
L22	*Thermus thermophilus*	X-ray
L30	*Bacillus stearothermophilus*	X-ray

Table 5.2: Ribosomal proteins whose structures have been determined [RW98]. The proteins within the small subunit are labelled with an "S", the proteins in the large subunit with an "L".

Pseudo-Atoms

Additional pseudo-atoms are generated and attached to the helix, in order to prevent single strands and other helices from penetrating the helix: V-type pseudo-atoms are placed between facing strands, in the middle of each hydrogen bonded base-pair. W-type pseudo-atoms are placed between stacked residues of the same strand, at half the distance of the C1′ atoms of two successive residues (figure 5.6).

5.3.3 Known Complexes

Complexes with a known structure, that have already been discussed in sections 4.3.2 and 4.3.3) are ideal to be modelled as rigid objects: Proteins (listed in table 5.2) can be clustered into rigid objects. They can then be allowed to move accordingly or they can be spatially fixed. For RNA segments or protein-RNA complexes only the appropriate nucleotides need to be specified in order for a rigid object to be generated.

In the model described in the following such objects have been generated for the *loop E-loop D region* of the 5S NMR structure [DM97], for the 23S *sarcin/ricin loop* NMR structure [SM95], for the *L11-RNA complex* [WGM+99], and for the decoding area of the 16S RNA [FRBP96, vLEH99].

5.4 Cross-Linking Data

The energy potential, defined in section 1.2, reflects structural data not only at the atomic level, but also at a coarser level of resolution. For the ribosome a set of distance measurements derived by cross-linking methods is incorporated.

There are a number of cross-linking data established for the ribosome, as discussed in section 4.2.5. A reagent is used to establish a link between certain residues, which can be used as a distance measurement in the modelling process. For this the cross-linking energy potential, defined by equation (1.4), is used. The length d_l depends on the reagent. For some reagents it is not clear to which atoms within the residues they bind. In such cases all atoms in the cross-linked residues are used for distance measurement and are included into the atom sets X_l or Y_l. For other reagents it is known to which atoms they bind – and only these atoms are included in the distance measurement. Tables of ribosomal cross-linking data are listed in appendix C. The listed protein-RNA cross-linking data were not used to modify (i.e. the according forces were switched off) but only to evaluate the model.

5.5 Electron Microscopy Density

Finally, the energy potential includes information on density. For the most recent modelling cycle, a 3D density of the 70S ribosome was used, that was generated with the cryo-electron microscopy technique at a resolution of 13 Ångstrom Holger (H. Stark, M. Rodnina, F. Zemlin, W. Wintermayer, and M. van Heel, submitted for publication). The original input data were given as 128 x 128 x 128 voxel elements, each specified as a byte value 0 – 255; The higher the number, the higher the density. Thus voxels with a density below a certain value can be considered "outside".

For visualization purposes this density information was converted into a polygon contour [Mue93, Erd96], as depicted in colour plates 3.5f, 6.2, and 6.3ff. It was scaled and oriented to be in the same coordinate system as the ribosomal RNA atom positions.

In order not to introduce artefacts into the computational process, the density was then pretreated by thresholding byte-values below 15 and above 50, and rescaling to the range 0 – 255.

Chapter 6

Modelling Process

6.1 Overview

The aim of this chapter is to describe the iterative flow of information that was applied at the *Max-Planck-Institut für molekulare Genetik* for finding a plausible structure of the ribosome.

This flow of information is schematically depicted in figure 6.1. Each box symbolizes a technique that the model is subjected to during the refinement process, until a suitable structure is found after several iterations.

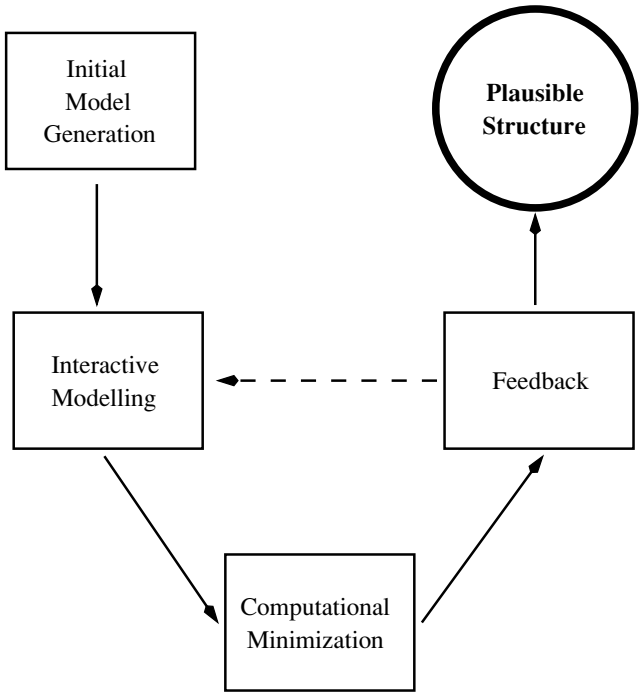


Figure 6.1: Flow of information for generating a plausible model of the ribosome

Whereas the individual techniques were described in detail before, their applicability including the data processing is elaborated in the following sections which are organized according to the diagram.

The arrows in the diagram symbolize the information carried through during iteration. Where the arrows are solid, the model in a stage of refinement is meant as information transport, where the arrows are dashed, either the model or feedback is the transported information.

6.2 Initial Model

An initial geometric model of the ribosomal RNA structures can be derived from the secondary structure maps as depicted in figures 5.1 to 5.3 [Mue93]. As mentioned in section 3.1.2, the letters A, C, G, U in the structure map corresponds to nucleotides. For each type of nucleotide a standard 3D structure is specified, as a set of coordinates for the atoms involved. These standard structures of the nucleotides reflect the constraints specified in appendix B. Analogously, default 3D structures for the standard helices are defined. By successively following the strands, inserting the standard structures for each nucleotide, and additionally respecting the helices the initial model is generated.

This initial model is based on geometric considerations only, and serves as a basis for further refinement. The next steps are to incorporate higher-level biochemical information in an interactive modelling process and take into account the principles of physics as here the principle of energy minimization.

6.3 Interactive Modelling

6.3.1 First Refinement of the Geometric Model

With the editing techniques discussed in section 2.3 on interactive manipulation methods, the helices of the geometric model are fitted into the cryo-electron microscopy density and placed in relative positions that are supported by the biochemical evidence gained by investigations exemplified in chapter 4. In order to meet geometric requirements single strands are moved so that they connect the adjoining helices – or are modelled specifically if supporting data exist. Where available, protein structures, are fitted onto the RNA, for an example see [TNH$^+$98].

In view of the size of the ribosome, this phase of modelling is the most time consuming. It was performed with the programme Erna3D [Mue93, MB97a, MB97b, MSvH$^+$97, MSB$^+$eda, MSB$^+$edb]. The programme Erna3D has a built-in and extendable set of constraints that allow rapid manipulation of large structures. Originally it was designed specifically for RNA, but in the meantime it has been extended to manipulate proteins.

6.3.2 Plausibility Checks

Each interactive manipulation step is accompanied by plausibility checks according to the energy model elaborated in this thesis. Cross-links are checked, and spatial areas that have been modelled in a sufficiently detailed manner are subjected to energy evaluation with the complete potential as specified in chapter 5. These checks result in graphical or textual output as described in section 2.1.1, to be used for iterative refinements of the model accomplished with Erna3D.

6.3.3 Artefacts of Interactive Manipulation

Interactive modelling on the basis of the programme Erna3D satisfies various aspects of the energy potential (1.12) to a different degree:

Bonds that do not participate in the backbone manipulation, i. e. those in the sugar rings and bases of single strand residues as well as those in helices, are mostly conserved from the initial geometric model. The corresponding bond lengths and bond angles are therefore close to the desired values. The backbone manipulation of Erna3D heavily modifies bond lengths and especially bond angles. These bond lengths usually range from somewhere between 0.5 to 4.0 Ångstrom, whereas they should be around 1.5 Å. Angles of bonds involved in the backbone range from arbitrary to optimal values.

The Van-der-Waals constraints are satisfied at the "large" scale level, but not at the "local" level: The modeller and feedback mechanisms ensure that two helices do not completely overlap and occupy the same space. However there might be regions of contact. Some single strands are modelled less precisely; they may intersect, resulting in immense energies.

Cross-linking constraints and the cryo-electron microscopy density are satisfied in general, since they play an integral role in placing the helices.

6.4 Computational Minimization

After the described interactive manipulation, the model satisfies the biochemical constraints such as cross-linking and foot-printing data to a high degree and it is considered to be "close" to a plausible, "deep" minimum of the energy potential. The closeness refers to the parameters, i. e. the atom positions, not to the actual energy value which is often outrageous.

In order to reduce the deficiencies described in the previous section, the model is subjected to energy minimization. For this the clustered molecular dynamics technique (cf. chapter 1) is used, in combination with a ribosome specific energy potential (cf. chapter 5).

Minimization is carried out in a stepwise manner, successively eliminating deficiencies from major to minor problems.

Minimization Step I

Adaptation of Lennard-Jones Potential: The modeller might have not been careful about a – probably biochemically unimportant – detail, and for instance may have modelled two nucleotides to occupy the same space. This yields a potentially infinite Van-der-Waals energy, resulting in high forces, accelerating the atoms involved so much that the model literally "explodes".

In order to avoid this effect, the Van-der-Waals energies are limited artificially by adaptation of the Lennard-Jones potential function as described in section 1.6.2.
CMD Parameters: For the ensuing clustered molecular dynamics run, the ribosomal RNA helices are clustered into objects, equipped with V- and W-type pseudo-atoms and held fixed. The single strands are clustered into residues and equipped

with Y-type pseudo-atoms. The CMD energy minimization technique is then used to reduce the potential function as specified in chapters 1.2 and 5 to a minimum.

Minimization Step II

Reestablishment of Lennard-Jones Potential: For this step the Lennard-Jones potential is set back to normal, as described by equation (1.84) in section 1.6.2.
CMD Parameters: For the next CMD run, the helices are held clustered, equipped with V- and W-type pseudo-atoms and fixed. The single strands are clustered into residues and equipped with X-type pseudo-atoms.
Result: The helix positions have so far not been changed. The single strands are connected to the helices, and can thus be forced into stereochemically impossible positions. Wherever this is not the case, the single strand residues do not overlap with other single strand or helix residues after this phase.

Minimization Step III

CMD Parameters: Another CMD run is started with the helices as before, and single strands clustered at the sub-residue level, still equipped with X-type pseudo-atoms.
Result: This phase improves the structural qualities of the single strands. Bond angles and bond lengths, especially in the backbone, are corrected.

Minimization Step IV

CMD Parameters: The helices are allowed to move, still clustered and equipped with V- and W-type pseudo-atoms. The residues are clustered at sub-residue level and equipped with X-type pseudo-atoms.
Result: Helices which were touching due to the interactive modelling are pushed apart from each other. Single strands follow, and keep their structural properties as far as possible.

Minimization Step V

CMD Parameters: Helices remain clustered as a whole, single strand residues are clustered at sub-residue level. All pseudo-atoms are removed.
Result: This removes potential artefacts that could have been introduced by pseudo-atoms pushing other atoms too far away.

Minimization Step VI

CMD Parameters: All rigid objects are declustered into an all-atomic model, the cross-link and the density functions are switched off from the energy potential, and a final MD minimization run is started.
Result: This has not been done with the current model [MSB^{+}eda, MSB^{+}edb], but assures optimal low-level structural properties by minimally moving individual atoms.

Step	Helices	Helix Pseudo-Atoms	Single Strand Clustering	Single Strand Pseudo-Atoms
I	clustered, fixed	V, W	residue level	Y
II	clustered, fixed	V, W	residue level	X
III	clustered, fixed	V, W	sub-residue level	X
IV	clustered	V, W	sub-residue level	X
V	clustered	—	sub-residue level	—
VI	unclustered	—	atom level	—

Table 6.1: Steps of the energy minimization process

6.5 Feedback and Results of the Refinement

When too many helices are placed in the same spatial area by the interactive manipulation process, plausible solutions for the structure generated can not be found automatically by the minimization process as described in the previous section. In this case, the programme indicates to the modeller that there is an impossible situation in this area. He thus has to find an alternative arrangement for the helices concerned, and this manual solution is refined by energy minimization anew. If the surgery is small scale, the model needs only partial recalculation, and the energy minimization can start at an advanced step.

The model resulting from this work is published elsewhere [MSB$^+$eda, MSB$^+$edb], here the focus is on the structural aspects. The general advantage of the methods discussed in this thesis is the physicochemical motivation of individual steps of the modelling process. The results of the developed techniques can best be seen by looking at the individual effects:

- The generated structural model is plausible, according to physicochemical demands, as specified by the energy potential function in section 1.2. For an example, see colour plate 6.1.
- Higher level constraints given by cryo-electron microscopy (cf. section 4.3.4) and cross-linking techniques (cf. section 4.2.5) are incorporated into the model, which is exemplified in plate 6.2.
- Modifications of the overall structural model are achieved, including and maintaining precise sub-structures, investigated in detail by other techniques. The sub-structures listed in section 5.3.3 were included into the above mentioned model.

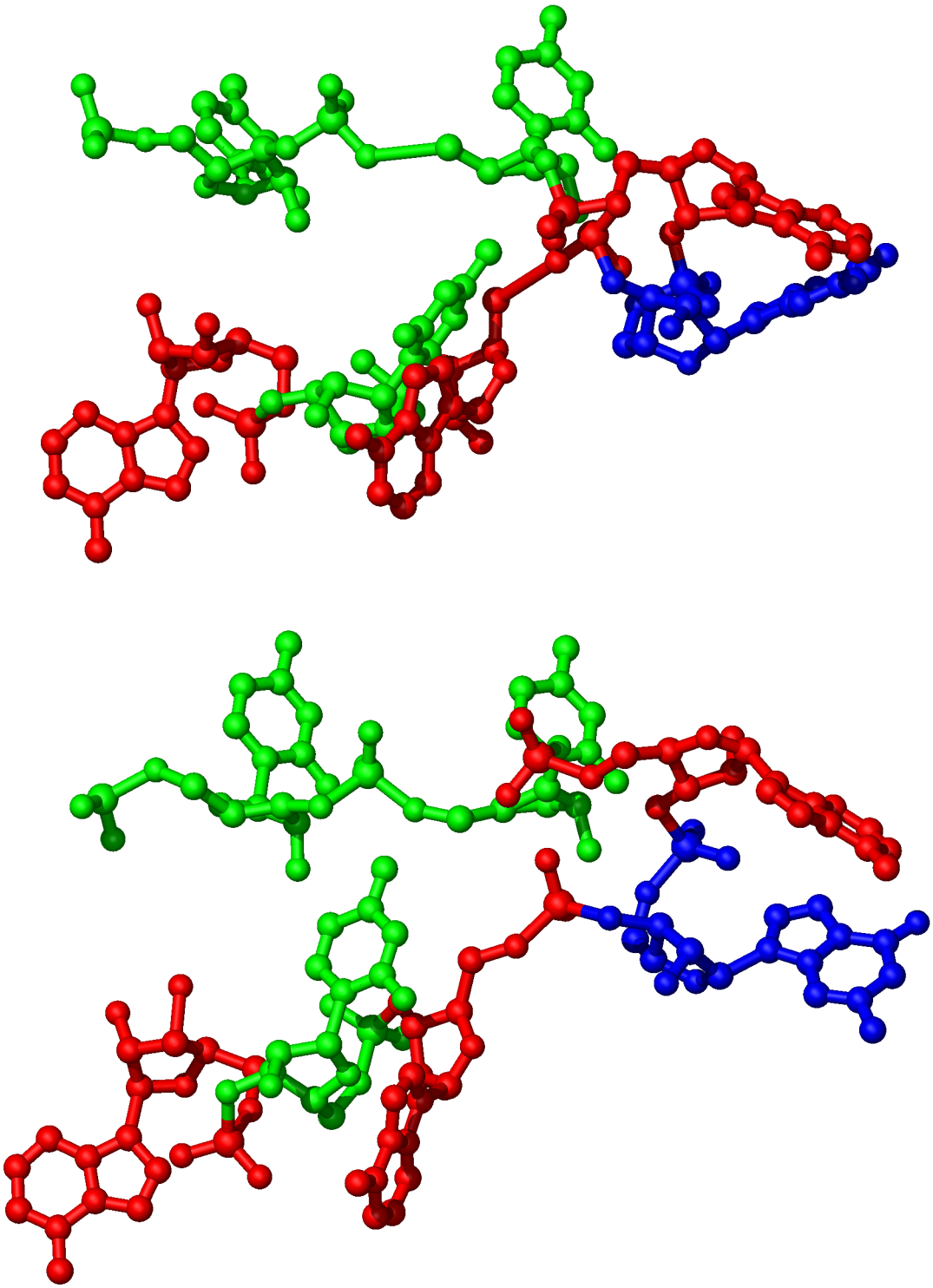


Plate 6.1: Residues 1044 – 1050 of the 23S rRNA. Top: after interactive placement with Erna3D; Bottom: as improved by the energy minimization process: Overlapping bases (e.g. on the right side) are pushed apart, bond lengths are corrected (e.g. above, middle), and bond angles are improved (e.g. lower right). The nucleotides are coloured as in plate 4.1.

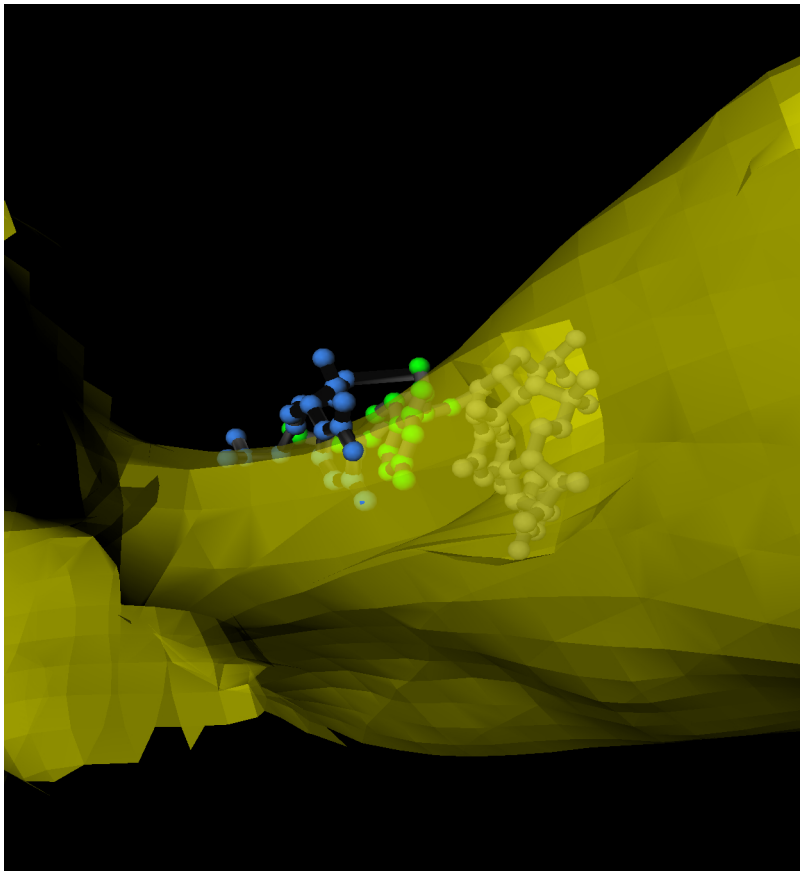


Plate 6.2: Residue 1364 of the small subunit as pushed into density by the energy minimization process: The residue as left by the interactive modelling (blue); the same residue after energy minimization (green) pushed into density (yellow). The grey residues within density belong to a helix that was spatially fixed and thus keeps the green residue from moving farther into the density.

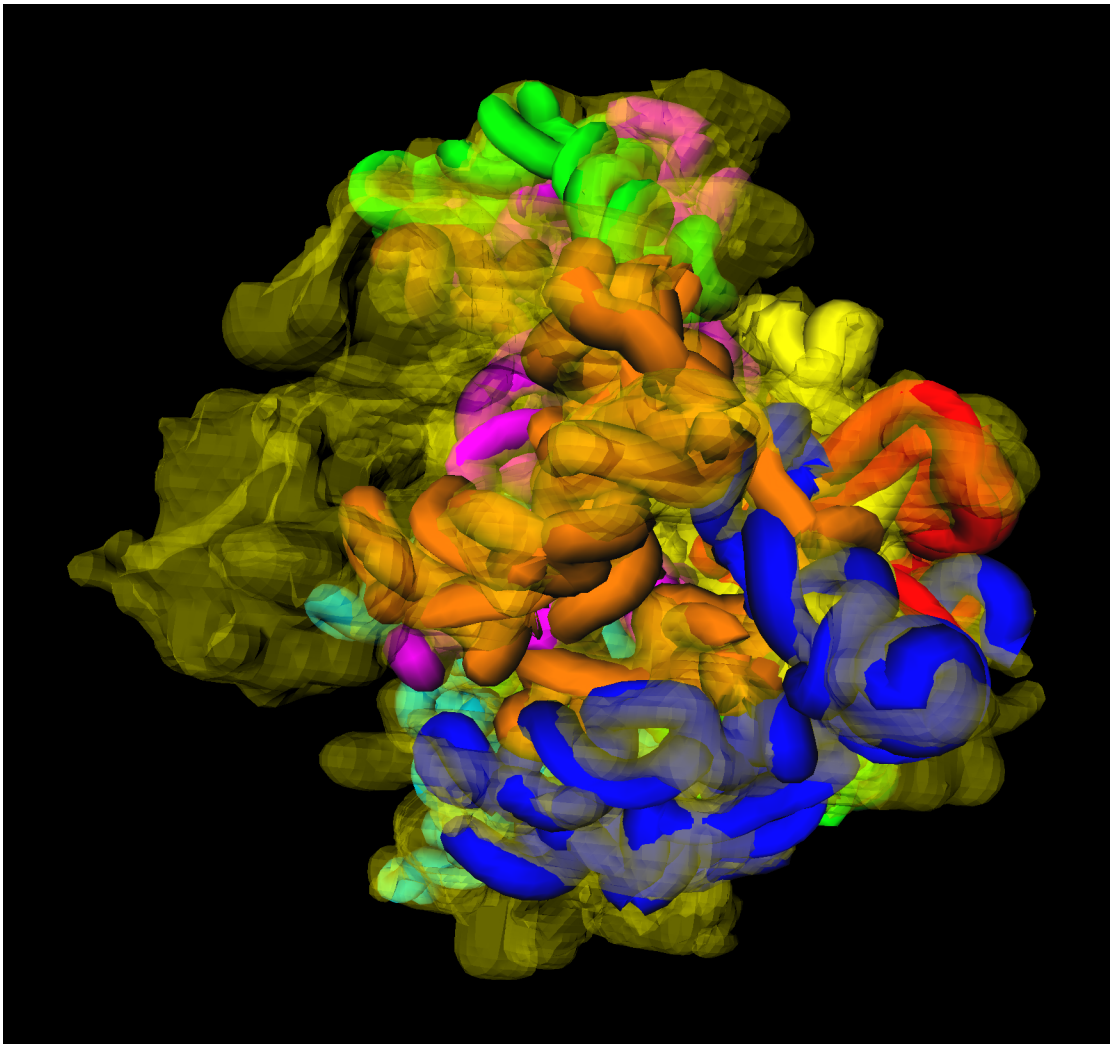


Plate 6.3: Large Subunit with 23S and 5S rRNA modelled into it. The rRNA is drawn as a thick tube model and colour coded according to domains (cf. figures 5.2 and 5.3): 23S I red, IIA yellow, IIB orange, III green, IV light blue, V purple, VI blue and 5S green

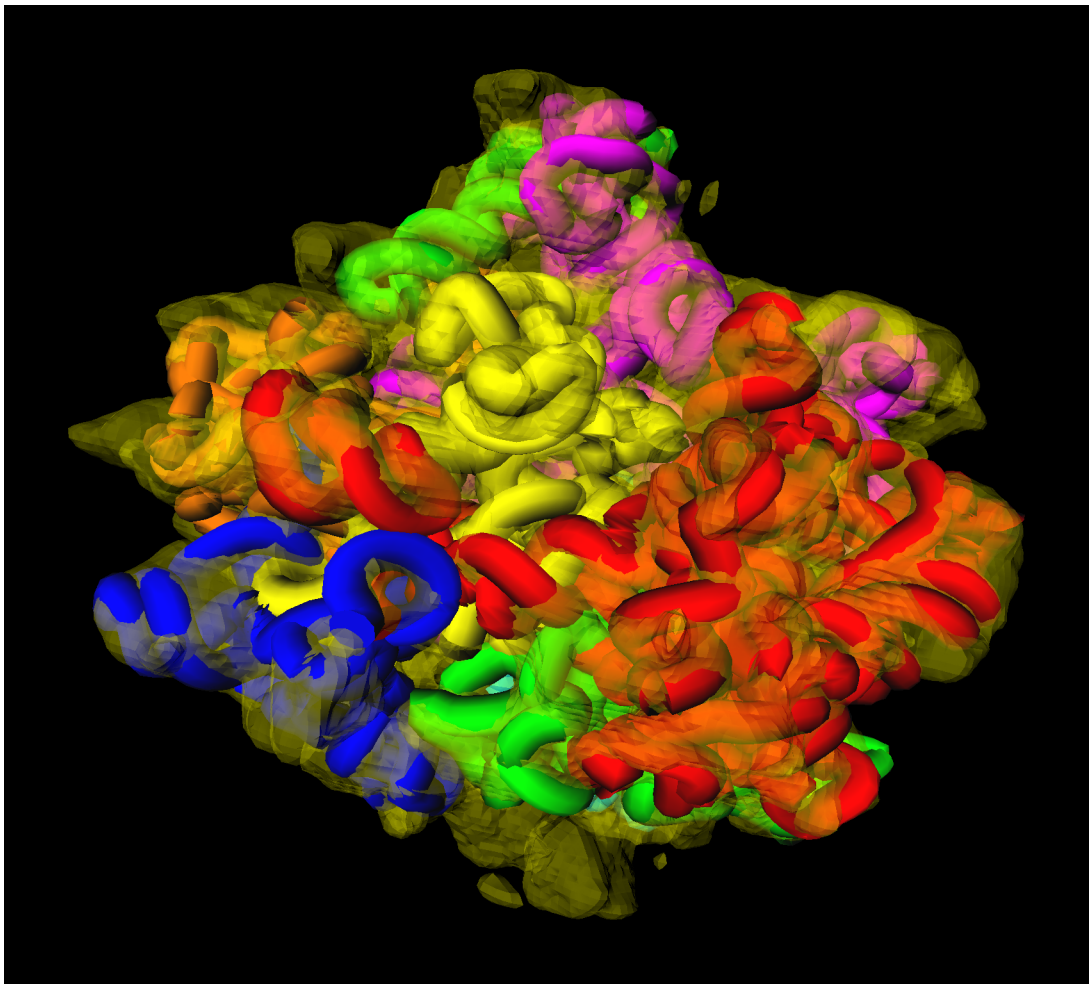


Plate 6.4: Large Subunit with 23S and 5S RNA modelled into it, colour coding as in plate 6.3, y-rotated -90 degrees

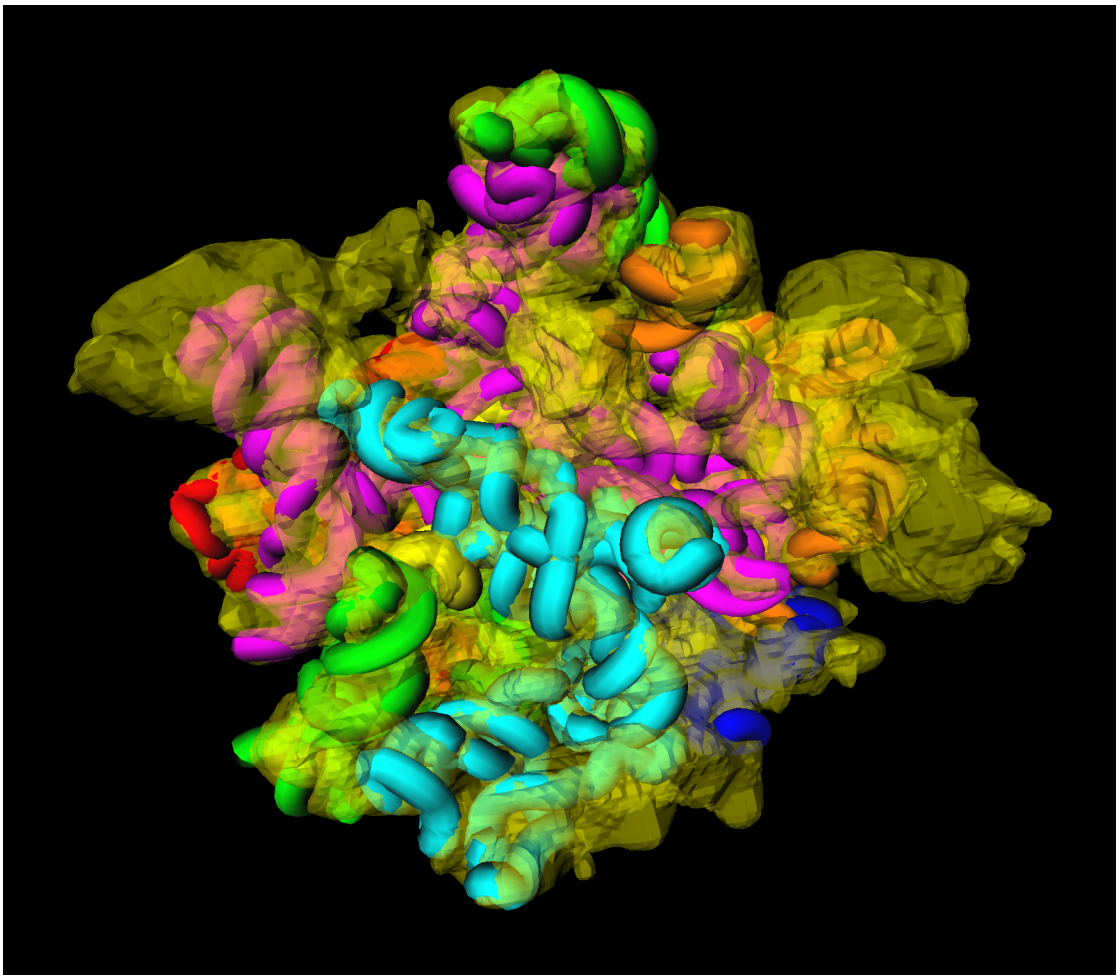


Plate 6.5: Large Subunit with 23S and 5S RNA modelled into it, colour coding as plate 6.3, y-rotated 90 degrees

Appendix A

Rotation in $\mathbb{R}^3$

For an introduction to matrices in the context of transforms of coordinate-systems see [FvDFH90]. The underlying mathematics of groups is described in any book on algebra, e.g. [Bos93, Lan93]. Vector spaces and isometries are found in books on linear algebra, e.g. [DW89, RT89]. Rigid body motion is summarized in [MLS94]. Here some facts that other sections rely on are summarized, and stated without proof.

A.1 Dotproduct and Crossproduct

Recall, that the dot product between two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ is defined as

$$\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^T \boldsymbol{b}, \tag{A.1}$$

and that the cross product between two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3$ is defined as

$$\boldsymbol{a} \times \boldsymbol{b} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \times \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{pmatrix}. \tag{A.2}$$

A.2 Matrices, $\mathbb{R}^{n \times n}$

A matrix $M_\Phi \in \mathbb{R}^{n \times n}$ corresponds to a linear map $\Phi : \mathbb{R}^n \to \mathbb{R}^n$; by applying matrix multiplication to a vector $\boldsymbol{a} \in \mathbb{R}^n$, a vector $\boldsymbol{b} \in \mathbb{R}^n$ is obtained:

$$\boldsymbol{b} = \Phi(\boldsymbol{a}) = M_\Phi \, \boldsymbol{a}. \tag{A.3}$$

A.3 Skew-Symmetric Matrices, $so(3)$

The set of $n \times n$ skew-symmetric matrices is defined as

$$so(n) = \{M \in \mathbb{R}^{n \times n} : M^T = -M\}. \tag{A.4}$$

The skew-symmetric matrices $so(n)$ form a vector space over the reals. A 3×3 skew-symmetric matrix $M \in so(3)$ generally has the form

$$M = \begin{bmatrix} 0 & -m_z & m_y \\ m_z & 0 & -m_x \\ -m_y & m_x & 0 \end{bmatrix} \quad \text{with } m_x, m_y, m_z \in \mathbb{R}. \tag{A.5}$$

Its application to a vector $\boldsymbol{b} \in \mathbb{R}^3$ can be represented by the cross product:

$$Mb = \begin{bmatrix} 0 & -m_z & m_y \\ m_z & 0 & -m_x \\ -m_y & m_x & 0 \end{bmatrix} \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} m_y b_z - m_z b_y \\ m_z b_x - m_x b_z \\ m_x b_y - m_y b_x \end{pmatrix} = \boldsymbol{m} \times \boldsymbol{b}. \tag{A.6}$$

A.4 Rotation Matrices, $SO(3)$

A matrix $M \in \mathbb{R}^{n \times n}$ is orthogonal, if and only if $MM^T = Id$, that is if $M^T = M^{-1}$. In this case the row vectors of M form an orthogonal basis of $\mathbb{R}^n$. Applied to vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$, an orthogonal matrix is compatible with the inner product:

$$M\boldsymbol{a} \cdot M\boldsymbol{b} = \boldsymbol{a} \cdot \boldsymbol{b}. \tag{A.7}$$

Interpreted geometrically, this means that mapping vectors with an orthogonal matrix M leaves angles and distances untouched.

An orthogonal matrix R with the additional property of $\det(R) = 1$ is called a special orthogonal matrix. These matrices form a group under the operation of matrix multiplication. In three dimensions, this group is labelled $SO(3)$ and referred to as the rotation group of $\mathbb{R}^3$. Mapping with $R \in SO(3)$ additionally preserves the handedness of the mapped vectors, excluding mirror-images, thus the cross product is preserved:

$$R\boldsymbol{a} \times R\boldsymbol{b} = R(\boldsymbol{a} \times \boldsymbol{b}). \tag{A.8}$$

Every coordinate frame that is free to rotate relative to a fixed frame can be identified with a unique rotation $R \in SO(3)$. A vector with coordinates $\boldsymbol{a}' \in \mathbb{R}^3$ in the rotated frame has coordinates $\boldsymbol{a} = R\boldsymbol{a}'$ in the fixed frame.

Defining three axes $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^3$, $\boldsymbol{x} = (1,0,0)^T$, $\boldsymbol{y} = (0,1,0)^T$, and $\boldsymbol{z} = (0,0,1)^T$, the matrix $R_x(\psi)$ that rotates any vector $\boldsymbol{a}$ by the angle ψ around the $\boldsymbol{x}$-axis can be written as follows:

$$R_x(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix}. \tag{A.9}$$

Analogously the matrices rotating around the $\boldsymbol{y}$ and $\boldsymbol{z}$ axes respecively are

$$R_y(\psi) = \begin{pmatrix} \cos\psi & 0 & \sin\psi \\ 0 & 1 & 0 \\ -\sin\psi & 0 & \cos\psi \end{pmatrix} \quad \text{and} \quad R_z(\psi) = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{A.10}$$

A.5 Euler Angles

Conventionally, especially in mechanics texts e.g. [JG95], rotations in $\mathbb{R}^3$ are represented by Euler angles [Hog92, MLS94]. They specify rotation as the composition of three independent rotations about given axes through an origin in a given order. Often used is the ZYZ system, which states an arbitrary rotation R as

$$\begin{aligned} R &= R_z(\alpha)R_y(\beta)R_z(\gamma) \\ &= \begin{bmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & -\cos\alpha\cos\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta \\ \sin\alpha\cos\beta\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\cos\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta \\ -\sin\beta\cos\gamma & \sin\beta\sin\gamma & \cos\beta \end{bmatrix} \end{aligned} \tag{A.11}$$

Every rotation $R \in SO(3)$ can be parameterized by three such angles, and equation (A.11) can be solved for α, β, and γ. But singularities in the parameterization occur, the angles are not unique for the identity rotation, for example, $R_z(\alpha) * R_y(0) * R_z(-\alpha) = Id$ for any α.

Other types of Euler angles use different ordered sets of rotation axes. Common choices include ZYX axes (yaw-pitch-roll angles) and YZX (Helmholtz angles). Both parameterizations have the advantage of not having singularities at the identity orientation $R = Id$, though they do contain singularities at other orientations.

A.6 Quaternions

The quaternions are an algebraic structure extending the idea of complex numbers [Ham43, Ham44, FvDFH90, Hog92, MLS94]. A quaternion $\tilde{q}$ is an expression of the form

$$\tilde{q} = q_0 + q_1\tilde{\imath} + q_2\tilde{\jmath} + q_3\tilde{k}, \quad \text{with } q_0, q_1, q_2, q_3 \in \mathbb{R}, \tag{A.12}$$

where q_0 is the scalar component of $\tilde{q}$ and $\boldsymbol{q} = (q_1, q_2, q_3)^T$ is the vector component. A convenient shorthand notation is $\tilde{q} = (q_0, \boldsymbol{q})$. The set of quaternions is a four-dimensional vector space over the reals and forms a group with respect to quaternion multiplication. Multiplication is distributive and associative, but not commutative. It satisfies the relations:

$$\tilde{\imath}^2 = \tilde{\jmath}^2 = \tilde{k}^2 = -1, \tag{A.13}$$

$$\tilde{\imath}\tilde{\jmath} = \tilde{k}, \quad \tilde{\jmath}\tilde{k} = \tilde{\imath}, \quad \tilde{k}\tilde{\imath} = \tilde{\jmath}, \tag{A.14}$$

$$\tilde{\jmath}\tilde{\imath} = -\tilde{k}, \quad \tilde{k}\tilde{\jmath} = -\tilde{\imath}, \quad \tilde{\imath}\tilde{k} = -\tilde{\jmath}. \tag{A.15}$$

Let $\tilde{q}' = q_0' + q_1'\tilde{\imath} + q_2'\tilde{\jmath} + q_3'\tilde{k} = (q_0', \boldsymbol{q}')$ be a second quaternion. With no linear equation between $\tilde{\imath}, \tilde{\jmath}, \tilde{k}$, the equation $\tilde{q}' = \tilde{q}$ implies, that $q_0' = q_0$, $q_1' = q_1$, $q_2' = q_2$, and $q_3' = q_3$. Addition of two quaternions happens element-wise

$$\begin{aligned} \tilde{q} + \tilde{q}' &= q_0 + q_0' + (q_1 + q_1')\tilde{\imath} + (q_2 + q_2')\tilde{\jmath} + (q_3 + q_3')\tilde{k} \\ &= (q_0 + q_0',\ \boldsymbol{q} + \boldsymbol{q}'), \end{aligned} \tag{A.16}$$

multiplication follows as

$$\begin{aligned}\tilde{q}\tilde{q}' &= (q_0q_0' - q_1q_1' - q_2q_2' - q_3q_3') + (q_0q_1' + q_1q_0' + q_2q_3' - q_3q_2')\tilde{\imath} \\ &\quad +(q_0q_1' + q_2q_1' + q_3q_0' + q_1q_3')\tilde{\jmath} + (q_0q_3' + q_3q_0' + q_1q_2' - q_2q_1')\tilde{k} \quad \text{(A.17)} \\ &= (q_0q_0' - \boldsymbol{q}\cdot\boldsymbol{q}',\ q_0\,\boldsymbol{q}' + q_0'\,\boldsymbol{q} + \boldsymbol{q}\times\boldsymbol{q}'). \quad \text{(A.18)}\end{aligned}$$

The conjugate of a quaternion is defined as $\tilde{q}^* = q_0 - q_1\tilde{\imath} - q_2\tilde{\jmath} - q_3\tilde{k} = (q_0,\ -\boldsymbol{q})$. A unit quaternions satisfies the constraint $q_0^2+q_1^2+q_2^2+q_3^2 = 1$. The unit quaternions form a subgroup of the quaternions with respect to quaternion multiplication, where $\tilde{q}^*$ is the inverse of $\tilde{q}$. A pure quaternion is a quaternion with $q_0 = 0$. A pure quaternion can be identified with its vectorial part, $\tilde{q} = \boldsymbol{q}$, in a somewhat sloppy notation[1]. With a little algebra one sees, that for any quaternion $\tilde{q}$ and any pure quaternion $\tilde{x}$ the product $\tilde{y} = \tilde{q}\tilde{x}\tilde{q}^* = \tilde{q}\boldsymbol{x}\tilde{q}^*$ is always a pure quaternion, since the q_0 component of $\tilde{y}$ reduces to 0. Therefore $\tilde{y} = \boldsymbol{y} = \tilde{q}\boldsymbol{x}\tilde{q}^*$ can be written as 3x3 matrix multiplication:

$$\boldsymbol{y} = M\boldsymbol{x} = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & -2q_0q_3+2q_1q_2 & 2q_0q_2+2q_1q_3 \\ 2q_0q_3+2q_1q_2 & q_0^2-q_1^2+q_2^2-q_3^2 & -2q_0q_1+2q_2q_3 \\ -2q_0q_2+2q_1q_3 & 2q_0q_1+2q_2q_3 & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \boldsymbol{x} \quad \text{(A.19)}$$

If $\tilde{q}$ is a unit quaternion, the matrix M can be veryfied to be a rotation matrix. Furthermore, rotating by ψ about the unit vector $\boldsymbol{v} = (q_1, q_2, q_3)^T$ corresponds to the quaternion

$$\begin{aligned}\tilde{q} &= \cos(\psi/2) + q_1\sin(\psi/2)\ \tilde{\imath} + q_2\sin(\psi/2)\ \tilde{\jmath} + q_3\sin(\psi/2)\ \tilde{k} \\ &= (\cos(\psi/2) + \sin(\psi/2),\ \boldsymbol{v}). \quad \text{(A.20)}\end{aligned}$$

Thus any rotation in three dimensional space can be represented as a quaternion. Since rotating by $-\psi$ about $-\boldsymbol{v}$ is the same as rotating by ψ about $\boldsymbol{v}$, each rotation in three-space corresponds to exactly two quaternions.

For any unit quaternion, the corresponding rotation matrix is given by (A.19). Finding the unit quaternion for a given rotation matrix is possible by solving this matrix-equation for q_0, q_1, q_2, and q_3.

[1] The notation $\boldsymbol{q} = \tilde{q}$ is used for pure quaternions only.

Appendix B

Nucleotide Patterns

B.1 Nucleotide Patterns

For each type of nucleotide a pattern is specified, that describes the properties of the nucleotide as discussed in chapter 5. Each nucleotide has a name "A" for adenylic acid, "C" for cytidylic acid, "G" for guanylic acid, and "U" for uridylic acid. The atoms within the residues are enumerated, and bonds, bond angles, and clustering groups are specified with respect to these numbers. The bonds are symmetric, thus

```
bond 1 2        length 1.546  // comment
```

defines the first and the second atom within the nucleotide to be covalently bond, the desired bond lenth to be 1.546 Å, and the force constant to be the programme default. Denoted by two dashed, the rest of the line is a comment.

The angles are non-symmetric, and

```
angle  1  2  3  deg 107    k 110   // comment
```

specifies the angle with atoms 2 and 3 at the ends and atom 1 in the middle (figure 1.2.2 on page 8 with $i = 1$, $j = 2$, $k = 3$). The desired angle is 107 degrees, and the force constant is 110.

Group definitions are slightly more powerful. The example

```
group 13 - 22
  pseudoatom X  13 - 16, 22       maxradius 1.5
  pseudoatom X  16, 17, 19 - 22   maxradius 1.5
```

specifies a base to be potentially clustered into a rigid object at the sub-residue level. The atoms $13, 14, \ldots 22$ form the complete base and are the group to be clustered. If the atoms are clustered, two X-type pseudoatoms are attached to the object. The position of each pseudoatom is defined to be the center of the atoms specified succesively. The radius of the pseudoatom is the maximum distance from one of the specified atoms to the center, scaled with constant specified after `maxradius`. Thus, the first pseudoatom is attached to the center of atoms 13, 14, 15, 16, 22 and its radius is scaled with 1.5.

In the sequel patterns for the standard nucleotides are listed. For nonstandard nucleotides within the ribosome, analogous patterns were defined.

B.2 Adenylic Acid Pattern

```
#MPIMG Molecular Pattern V1.0 ascii
// $ Id: A.pat,v 1.5 1999/01/20 16:15:44 sommer Exp sommer $

residue A

// specifying numbers is optional
// if not specified, first atom is atom number one
atom {
 0  OXT or lastResidue
 1  P
 2  O1P
 3  O2P
 4  "O5*"
 5  "C5*"
 6  "C4*"
 7  "O4*"
 8  "C3*"
 9  "O3*"
10  "C2*"
11  "O2*"
12  "C1*"
13  N9
14  C8
15  N7
16  C5
17  C6
18  N6
19  N1
20  C2
21  N3
22  C4
23 nextResidue
}
bond 0 1        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 1 2        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 1 3        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 1 4        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 4 5        length 1.42   // Saenger, p. 70   (C2' - endo)
bond 5 6        length 1.514  // Saenger, p. 70
bond 6 7        length 1.453  // Saenger, p. 70
bond 6 8        length 1.526  // Saenger, p. 70
bond 7 12       length 1.415  // Saenger, p. 70
bond 8 9        length 1.424  // Saenger, p. 70
bond 9 23       length 1.546  // Saenger, p. 83  == bond  0 1
bond 8 10       length 1.528  // Saenger, p. 70
bond 10 11      length 1.409  // Saenger, p. 70
bond 10 12      length 1.526  // Saenger, p. 70

bond 12 13      length 1.479  // Saenger, p. 52   (averaged)
bond 13 14      length 1.367  // Saenger, p. 52
bond 13 22      length 1.376  // Saenger, p. 52
bond 14 15      length 1.312  // Saenger, p. 52
bond 15 16      length 1.385  // Saenger, p. 52
bond 16 17      length 1.409  // Saenger, p. 52
bond 16 22      length 1.382  // Saenger, p. 52
bond 17 18      length 1.337  // Saenger, p. 52
bond 17 19      length 1.349  // Saenger, p. 52
bond 19 20      length 1.338  // Saenger, p. 52
bond 20 21      length 1.332  // Saenger, p. 52
bond 21 22      length 1.342  // Saenger, p. 52

// four base angles are most important
angle  1  0  2  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  0  3  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  0  4  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  2  3  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  2  4  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  3  4  deg 107    k 110   // Saenger, p. 84 (rough est.)
```

```
angle  5  4  6  deg 111.8  k 110   // Saenger, p. 70 (C2'- endo)
angle  6  5  7  deg 108.7  k 110   // Saenger, p. 70
angle  6  5  8  deg 114.9  k 110   // Saenger, p. 70
angle  6  7  8  deg 106.4  k 110   // Saenger, p. 70
angle  7  6 12  deg 109.1  k 110   // Saenger, p. 70
angle  8  6  9  deg 109.4  k 110   // Saenger, p. 70
angle  8  6 10  deg 102.5  k 110   // Saenger, p. 70
angle  8  9 10  deg 110.0  k 110   // Saenger, p. 70
angle  9  8 23  deg 107.0  k 110   // Saenger, p. 84 (rough est.)
angle 10  8 11  deg 114.3  k 110   // Saenger, p. 70
angle 10  8 12  deg 101.1  k 110   // Saenger, p. 70
angle 10 11 12  deg 112.7  k 110   // Saenger, p. 70
angle 12  7 10  deg 105.6  k 110   // Saenger, p. 70
angle 12  7 13  deg 108.3  k 1100   // Saenger, p. 70
angle 12 10 13  deg 114.6  k 1100   // Saenger, p. 70

angle 13 12 14  deg 128.4  k 1100   // Saenger, p. 52
angle 13 12 22  deg 126.1  k 1100   // Saenger, p. 52
angle 13 14 22  deg 105.9  k 110   // Saenger, p. 52
angle 14 13 15  deg 113.8  k 110   // Saenger, p. 52
angle 15 14 16  deg 103.9  k 110   // Saenger, p. 52
angle 16 15 17  deg 132.3  k 110   // Saenger, p. 52
angle 16 15 22  deg 110.7  k 110   // Saenger, p. 52
angle 16 17 22  deg 116.9  k 110   // Saenger, p. 52
angle 17 16 18  deg 123.4  k 110   // Saenger, p. 52
angle 17 16 19  deg 117.6  k 110   // Saenger, p. 52
angle 17 18 19  deg 119.0  k 110   // Saenger, p. 52
angle 19 17 20  deg 118.8  k 110   // Saenger, p. 52
angle 20 19 21  deg 129.0  k 110   // Saenger, p. 52
angle 21 20 22  deg 110.8  k 110   // Saenger, p. 52
angle 22 13 16  deg 105.7  k 110   // Saenger, p. 52
angle 22 13 21  deg 127.4  k 110   // Saenger, p. 52
angle 22 16 21  deg 126.9  k 110   // Saenger, p. 52


group 0 - 3   // which does the 0 if it is OXT only
  pseudoatom X  maxradius 1.75  // factor
group 6 - 8, 10 - 12
  pseudoatom X  6 - 8, 10, 12 maxradius 2.0
  pseudoatom Y  6 - 7  maxradius 12.0
group 13 - 22
  pseudoatom X  13 - 16, 22       maxradius 1.5
  pseudoatom X  16, 17, 19 - 22   maxradius 1.5
```

B.3 Cytidylic Acid Pattern

```
#MPIMG Molecular Pattern V1.0 ascii
// $ Id: C.pat,v 1.5 1999/01/20 16:15:44 sommer Exp sommer $

residue C

atom {
 0  OXT or lastResidue
 1  P
 2  O1P
 3  O2P
 4  "O5*"
 5  "C5*"
 6  "C4*"
 7  "O4*"
 8  "C3*"
 9  "O3*"
10  "C2*"
11  "O2*"
12  "C1*"
13  N1
```

```
14  C2
15  O2
16  N3
17  C4
18  N4
19  C5
20  C6
21  nextResidue
}

bond 0 1        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 1 2        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 1 3        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 1 4        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 4 5        length 1.42   // Saenger, p. 70   (C2' - endo)
bond 5 6        length 1.514  // Saenger, p. 70
bond 6 7        length 1.453  // Saenger, p. 70
bond 6 8        length 1.526  // Saenger, p. 70
bond 7 12       length 1.415  // Saenger, p. 70
bond 8 9        length 1.424  // Saenger, p. 70
bond 9 21       length 1.546  // Sanger, p. 83  == bond  0 1
bond 8 10       length 1.528  // Saenger, p. 70
bond 10 11      length 1.409  // Saenger, p. 70
bond 10 12      length 1.526  // Saenger, p. 70
bond 12 13      length 1.468  // Saenger, p. 52   (averaged)
bond 13 14      length 1.339  // Saenger, p. 52
bond 13 20      length 1.364  // Saenger, p. 52
bond 14 15      length 1.237  // Saenger, p. 52
bond 14 16      length 1.356  // Saenger, p. 52
bond 16 17      length 1.334  // Saenger, p. 52
bond 17 18      length 1.337  // Saenger, p. 52
bond 17 19      length 1.426  // Saenger, p. 52
bond 19 20      length 1.337  // Saenger, p. 52

// four base angles are most important
angle  1  0  2  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  0  3  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  0  4  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  2  3  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  2  4  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  3  4  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  5  4  6  deg 111.8  k 110   // Saenger, p. 70 (C2'- endo)
angle  6  5  7  deg 108.7  k 110   // Saenger, p. 70
angle  6  5  8  deg 114.9  k 110   // Saenger, p. 70
angle  6  7  8  deg 106.4  k 110   // Saenger, p. 70
angle  7  6 12  deg 109.1  k 110   // Saenger, p. 70
angle  8  6  9  deg 109.4  k 110   // Saenger, p. 70
angle  8  6 10  deg 102.5  k 110   // Saenger, p. 70
angle  8  9 10  deg 110.0  k 110   // Saenger, p. 70
angle  9  8 21  deg 107.0  k 110   // Saenger, p. 84 (rough est.)
angle 10  8 11  deg 114.3  k 110   // Saenger, p. 70
angle 10  8 12  deg 101.1  k 110   // Saenger, p. 70
angle 10 11 12  deg 112.7  k 110   // Saenger, p. 70
angle 12  7 10  deg 105.6  k 110   // Saenger, p. 70
angle 12  7 13  deg 108.3  k 1100   // Saenger, p. 70
angle 12 10 13  deg 114.6  k 1100   // Saenger, p. 70

angle 13 12 14  deg 118.2  k 1100   // Saenger, p. 52
angle 13 12 20  deg 121.5  k 1100   // Saenger, p. 52
angle 14 13 15  deg 119.2  k 110   // Saenger, p. 52
angle 14 13 16  deg 118.9  k 110   // Saenger, p. 52
angle 14 15 16  deg 121.9  k 110   // Saenger, p. 52
angle 16 14 17  deg 120.0  k 110   // Saenger, p. 52
angle 17 16 18  deg 117.9  k 110   // Saenger, p. 52
angle 17 16 19  deg 121.8  k 110   // Saenger, p. 52
angle 17 18 19  deg 121.8  k 110   // Saenger, p. 52
angle 19 17 20  deg 117.6  k 110   // Saenger, p. 52
angle 20 13 19  deg 121.0  k 110   // Saenger, p. 52

group 0 - 3   // which does the 0 if it is OXT only
```

```
  pseudoatom X  maxradius 1.75  // factor
group 6 - 8, 10 - 12
  pseudoatom X  6 - 8, 10, 12 maxradius 2.0
  pseudoatom Y  6 - 7  maxradius 12.0
group 13 - 20
  pseudoatom X  13, 14, 16, 17, 19, 20 maxradius 1.75
```

B.4 Guanylic Acid Pattern

```
#MPIMG Molecular Pattern V1.0 ascii
// $ Id: G.pat,v 1.5 1999/01/20 16:15:44 sommer Exp sommer $


residue G

atom {
 0  OXT or lastResidue
 1  P
 2  O1P
 3  O2P
 4  "O5*"
 5  "C5*"
 6  "C4*"
 7  "O4*"
 8  "C3*"
 9  "O3*"
10  "C2*"
11  "O2*"
12  "C1*"
13  N9
14  C8
15  N7
16  C5
17  C6
18  O6
19  N1
20  C2
21  N2
22  N3
23  C4
24  nextResidue
}

bond 0 1        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 1 2        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 1 3        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 1 4        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 4 5        length 1.42   // Saenger, p. 70   (C2' - endo)
bond 5 6        length 1.514  // Saenger, p. 70
bond 6 7        length 1.453  // Saenger, p. 70
bond 6 8        length 1.526  // Saenger, p. 70
bond 7 12       length 1.415  // Saenger, p. 70
bond 8 9        length 1.424  // Saenger, p. 70
bond 9 24       length 1.546  // Sanger, p. 83    == bond 0 1
bond 8 10       length 1.528  // Saenger, p. 70
bond 10 11      length 1.409  // Saenger, p. 70
bond 10 12      length 1.526  // Saenger, p. 70
bond 12 13      length 1.476  // Saenger, p. 52
bond 13 14      length 1.374  // Saenger, p.52
bond 13 23      length 1.377  // Saenger, p.52
bond 14 15      length 1.304  // Saenger, p.52
bond 15 16      length 1.389  // Saenger, p.52
bond 16 17      length 1.415  // Saenger, p.52
bond 16 23      length 1.377  // Saenger, p.52
bond 17 18      length 1.239  // Saenger, p.52
bond 17 19      length 1.393  // Saenger, p.52
bond 19 20      length 1.375  // Saenger, p.52
```

```
bond 20 21      length 1.341  // Saenger, p.52
bond 20 22      length 1.327  // Saenger, p.52
bond 22 23      length 1.355  // Saenger, p.52

// four base angles are most important
angle  1  0  2  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  0  3  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  0  4  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  2  3  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  2  4  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  3  4  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  5  4  6  deg 111.8  k 110   // Saenger, p. 70 (C2'- endo)
angle  6  5  7  deg 108.7  k 110   // Saenger, p. 70
angle  6  5  8  deg 114.9  k 110   // Saenger, p. 70
angle  6  7  8  deg 106.4  k 110   // Saenger, p. 70
angle  7  6 12  deg 109.1  k 110   // Saenger, p. 70
angle  8  6  9  deg 109.4  k 110   // Saenger, p. 70
angle  8  6 10  deg 102.5  k 110   // Saenger, p. 70
angle  8  9 10  deg 110.0  k 110   // Saenger, p. 70
angle  9  8 24  deg 107.0  k 110   // Saenger, p. 84 (rough est.)
angle 10  8 11  deg 114.3  k 110   // Saenger, p. 70
angle 10  8 12  deg 101.1  k 110   // Saenger, p. 70
angle 10 11 12  deg 112.7  k 110   // Saenger, p. 70
angle 12  7 10  deg 105.6  k 110   // Saenger, p. 70
angle 12  7 13  deg 108.3  k 1100   // Saenger, p. 70
angle 12 10 13  deg 114.6  k 1100   // Saenger, p. 70

angle 13 12 14  deg 129.2  k 1100   // Saenger, p.52
angle 13 12 23  deg 125.5  k 1100   // Saenger, p.52
angle 13 14 23  deg 106.0  k 110   // Saenger, p.52
angle 14 13 15  deg 113.5  k 110   // Saenger, p.52
angle 15 14 16  deg 104.2  k 110   // Saenger, p.52
angle 16 15 17  deg 130.1  k 110   // Saenger, p.52
angle 16 15 23  deg 110.8  k 110   // Saenger, p.52
angle 16 17 23  deg 119.1  k 110   // Saenger, p.52
angle 17 16 18  deg 128.3  k 110   // Saenger, p.52
angle 17 16 19  deg 111.7  k 110   // Saenger, p.52
angle 17 18 19  deg 120.0  k 110   // Saenger, p.52
angle 19 17 20  deg 124.9  k 110   // Saenger, p.52
angle 20 19 21  deg 116.3  k 110   // Saenger, p.52
angle 20 19 22  deg 124.0  k 110   // Saenger, p.52
angle 20 21 22  deg 119.7  k 110   // Saenger, p.52
angle 22 20 23  deg 111.8  k 110   // Saenger, p.52
angle 23 13 16  deg 105.6  k 110   // Saenger, p.52
angle 23 13 22  deg 126.0  k 110   // Saenger, p.52
angle 23 16 22  deg 128.4  k 110   // Saenger, p.52


group 0 - 3   // which does the 0 if it is OXT only
   pseudoatom X  maxradius 1.75  // factor
group 6 - 8, 10 - 12
   pseudoatom X  6 - 8, 10, 12 maxradius 2.0  // factor
   pseudoatom Y  6 - 7  maxradius 12.0
group 13 - 23
   pseudoatom X  13 - 16, 23 maxradius 2.0  // factor
   pseudoatom X  16, 17, 19, 20, 22, 23 maxradius 2.0  // factor
```

B.5 Uridylic Acid Pattern

```
#MPIMG Molecular Pattern V1.0 ascii
// $ Id: U.pat,v 1.5 1999/01/20 16:15:44 sommer Exp sommer $

residue U


atom {
 0  OXT or lastResidue
```

```
 1  P
 2  O1P
 3  O2P
 4  "O5*"
 5  "C5*"
 6  "C4*"
 7  "O4*"
 8  "C3*"
 9  "O3*"
10  "C2*"
11  "O2*"
12  "C1*"
13  N1
14  C2
15  O2
16  N3
17  C4
18  O4
19  C5
20  C6
21  nextResidue
}


bond 0 1        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 1 2        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 1 3        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 1 4        length 1.546  // Sanger, p. 83    ( P-O bond, Text )
bond 4 5        length 1.42   // Saenger, p. 70   (C2' - endo)
bond 5 6        length 1.514  // Saenger, p. 70
bond 6 7        length 1.453  // Saenger, p. 70
bond 6 8        length 1.526  // Saenger, p. 70
bond 7 12       length 1.415  // Saenger, p. 70
bond 8 9        length 1.424  // Saenger, p. 70
bond 9 21       length 1.546  // Sanger, p. 83  == bond  0 1
bond 8 10       length 1.528  // Saenger, p. 70
bond 10 11      length 1.409  // Saenger, p. 70
bond 10 12      length 1.526  // Saenger, p. 70

bond 12 13      length 1.476  // Saenger, p. 52
bond 13 14      length 1.379  // Saenger, p. 52
bond 13 20      length 1.380  // Saenger, p. 52
bond 14 15      length 1.218  // Saenger, p. 52
bond 14 16      length 1.373  // Saenger, p. 52
bond 16 17      length 1.383  // Saenger, p. 52
bond 17 18      length 1.227  // Saenger, p. 52
bond 17 19      length 1.440  // Saenger, p. 52
bond 19 20      length 1.338  // Saenger, p. 52

// four base angles are most important
angle  1  0  2  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  0  3  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  0  4  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  2  3  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  2  4  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  1  3  4  deg 107    k 110   // Saenger, p. 84 (rough est.)
angle  5  4  6  deg 111.8  k 110   // Saenger, p. 70 (C2'- endo)
angle  6  5  7  deg 108.7  k 110   // Saenger, p. 70
angle  6  5  8  deg 114.9  k 110   // Saenger, p. 70
angle  6  7  8  deg 106.4  k 110   // Saenger, p. 70
angle  7  6 12  deg 109.1  k 110   // Saenger, p. 70
angle  8  6  9  deg 109.4  k 110   // Saenger, p. 70
angle  8  6 10  deg 102.5  k 110   // Saenger, p. 70
angle  8  9 10  deg 110.0  k 110   // Saenger, p. 70
angle  9  8 21  deg 107.0  k 110   // Saenger, p. 84 (rough est.)
angle 10  8 11  deg 114.3  k 110   // Saenger, p. 70
angle 10  8 12  deg 101.1  k 110   // Saenger, p. 70
angle 10 11 12  deg 112.7  k 110   // Saenger, p. 70
angle 12  7 10  deg 105.6  k 110   // Saenger, p. 70
angle 12  7 13  deg 108.3  k 1100   // Saenger, p. 70
```

```
angle 12 10 13  deg 114.6  k 1100   // Saenger, p. 70

angle 13 12 14  deg 117.1  k 1100   // Saenger, p. 52
angle 13 12 20  deg 120.9  k 1100   // Saenger, p. 52
angle 13 14 20  deg 121.3  k 110   // Saenger, p. 52
angle 14 13 15  deg 123.2  k 110   // Saenger, p. 52
angle 14 13 16  deg 114.8  k 110   // Saenger, p. 52
angle 14 15 16  deg 122.0  k 110   // Saenger, p. 52
angle 16 14 17  deg 127.0  k 110   // Saenger, p. 52
angle 17 16 18  deg 119.8  k 110   // Saenger, p. 52
angle 17 16 19  deg 114.7  k 110   // Saenger, p. 52
angle 17 18 19  deg 125.4  k 110   // Saenger, p. 52
angle 19 17 20  deg 119.2  k 110   // Saenger, p. 52
angle 20 13 19  deg 122.8  k 110   // Saenger, p. 52

group 0 - 3   // which does the 0 if it is OXT only
  pseudoatom X  maxradius 1.75  // factor
group 6 - 8, 10 - 12
  pseudoatom X  6 - 8, 10, 12 maxradius 2.0  // factor
  pseudoatom Y  6 - 7  maxradius 12.0
group 13 - 20
  pseudoatom X  13, 14, 16, 17, 19, 20  maxradius 1.5  // factor
```

Appendix C

Ribosomal Cross-Links

The cross-links listed in the following tables were used for the compuationally modelling the ribosomal structure. These are an excerpt from the database of ribosomal cross-links at the *Max-Planck-Institut für molekulare Genetik* (see section 4.2.5).

```
crosslink   A   31             WITH   A   48              length=3.0  k=100     "16S 31 w. 16S 48; Atmadja, Stiege"
crosslink   A   31 "N7"        WITH   A  306        "N7"  length=8.0  k=100     "16S 31 w. 16S 306; Atmadja"
crosslink   A   46 "N7"        WITH   A  362 -  364 "N7"  length=8.0  k=100     "16S 46 w. 16S 362- 364; Atmadja"
crosslink   A   54             WITH   A  353              length=3.0  k=100     "16S 54 w. 16S 353; Wilms"
crosslink   A  118             WITH   A  288 -  289       length=3.0  k=100     "16S 118 w. 16S 288- 289; Stiege"
crosslink   A  128 -  131 "N7" WITH   A  232        "N7"  length=8.0  k=100     "16S 128- 131 w. 16S 232; Atmadja"
crosslink   A  148 -  149 "N7" WITH   A  174        "N7"  length=8.0  k=100     "16S 148- 149 w. 16S 174; Atmadja"
crosslink   A  244             WITH   A  894              length=3.0  k=100     "16S 244 w. 16S 894; Wilms, refinement of Stiege"
crosslink   A  250        "N7" WITH   A  274 -  275 "N7"  length=8.0  k=100     "16S 250 w. 16S 274- 275; Atmadja"
crosslink   A  366 -  369      WITH   A  397 -  399       length=3.0  k=100     "16S 366- 369 w. 16S 397- 399; Stiege"
crosslink   A  441             WITH   A  494              length=3.0  k=100     "16S 441 w. 16S 494; Wilms"
crosslink   A  497             WITH   A  545 -  548       length=3.0  k=100     "16S 497 w. 16S 545- 548"
crosslink   A  562             WITH   A  884              length=3.0  k=100     "16S 562 w. 16S 884; Wilms, refinement of Stiege"
crosslink   A  561 -  562      WITH   A  788 -  789       length=25   k=100     "16S 561- 562 w. 16S 788-789; Mundus, Wollenzien"
crosslink   A  582             WITH   A  760              length=3.0  k=100     "16S 582 w. 16S 760; Wilms, refinement of Stiege"
crosslink   A  693 - 696 "N7"  WITH   A  794 ,  799 "N7"  length=8.0  k=100     "16S 693- 696 w. 16S 794, 799; Atmadja"
crosslink   A  693             WITH   A  788 -  789       length=25   k=100     "16S 693 w. 16S 788- 789; Mundus, Wollenzien"
crosslink   A  723             WITH   A  788 -  789       length=25   k=100     "16S 723 w. 16S 788- 789; Mundus, Wollenzien"
crosslink   A  779 -  785      WITH   A  801 -  803       length=3.0  k=100     "16S 779- 785 w. 16S 801 - 803; Stiege"
crosslink   A  788 - 789       WITH   A  866              length=25   k=100     "16S 788- 789 w. 16S 866; Mundus, Wollenzien"
crosslink   A  788 - 789       WITH   A  920 - 921        length=25   k=100     "16S 788- 789 w. 16S 920- 921; Mundus, Wollenzien"
crosslink   A  894             WITH   A 1468              length=3.0  k=100     "16S 894 w. 16S 1468; Wilms"
crosslink   A  934             WITH   A 1345              length=3.0  k=100     "16S 934 w. 16S 1345; Wilms, refinement of Stiege"
crosslink   A  967             WITH   A 1400              length=3.0  k=100     "16S 967 w. 16S 1400; Wilms"
crosslink   A  977 -  987      WITH   A 1216 - 1220       length=3.0  k=100     "16S 977- 987 w. 16S 1216- 1220; Stiege"
crosslink   A  991             WITH   A 1212              length=3.0  k=100     "16S 991 w. 16S 1212; Wilms, refinement of Stiege"
crosslink   A 1052             WITH   A 1200              length=3.0  k=100     "16S 1052 w. 16S 1200; Wilms"
crosslink   A 1065 - 1068      WITH   A 1188 - 1190       length=3.0  k=100     "16S 1065- 1068 w. 16S 1188- 1190"
crosslink   A 1090             WITH   A 1161 - 1164       length=3.0  k=100     "16S 1090 w. 16S 1161- 1164; Stiege, Stiege, Atmadja"
crosslink   A 1093             WITH   A 1182              length=3.0  k=100     "16S 1093 w. 16S 1182; Wilms"
crosslink   A 1126             WITH   A 1281              length=3.0  k=100     "16S 1126 w. 16S 1281; Wilms, refinement of Stiege"
crosslink   A 1238 - 1240      WITH   A 1298              length=3.0  k=100     "16S 1238- 1240 w. 16S 1298; Stiege, Atmadja"
crosslink   A 1348 - 1353      WITH   A 1374 - 1379       length=3.0  k=100     "16S 1348- 1353 w. 16S 1374- 1379; Stiege"
crosslink   A 1393 - 1401      WITH   A 1502 - 1504       length=3.0  k=100     "16S 1393- 1401 w. 16S 1502- 1504; Doring"
crosslink   A 1393 - 1401      WITH   A 1506              length=3.0  k=100     "16S 1393- 1401 w. 16S 1506; Doring"
crosslink   A 1393 - 1401      WITH   A 1531 - 1532       length=3.0  k=100     "16S 1393- 1401 w. 16S 1531- 1532; Doring"
crosslink   A  1402            WITH   A 1501              length=3.0  k=100     "16S 1402 w. 16S 1501; Wilms, refinement of Doring"
crosslink   A 1413        "N7" WITH   A 1486        "N7"  length=8.0  k=100     "16S 1413 w. 16S 1486; Atmadja"
crosslink   A 1432             WITH   A 1465 - 1469       length=3.0  k=100     "16S 1432 w. 16S 1465-1469; Doring"
```

Table C.1: 16S Cross-Links

```
crosslink   B  292 -  296      WITH   B  339 -  350        length=3.0  k=100     "23S 292- 296 w. 23S 339- 350; Stiege et al."
crosslink   B  571 -  577      WITH   B 2030 - 2032        length=3.0  k=100     "23S 571- 577 w. 23S 2030- 2032; Stiege et al."
crosslink   B  601 -  604      WITH   B  652 -  656        length=3.0  k=100     "23S 601- 604 w. 23S 652- 656; Stiege et al."
crosslink   B  739 -  748      WITH   B 2069 - 2618        length=3.0  k=100     "23S 739- 747 w. 23S 2069- 2618; Stiege et al."
crosslink   B  763        "N7" WITH   B 1567         "N7"  length=8.0  k=100     "23S 763 w. 23S 1567; Stiege et al."
crosslink   B  774 -  778 "N7" WITH   B  792 -  794  "N7"  length=8.0  k=100     "23S 774- 778 w. 23S 792- 794; Doring et al."
crosslink   B  867             WITH   B  913                length=23.0 k=100     "23S 867 w. 23S 913; Turner & Noller"
crosslink   B  876 -  879 "N7" WITH   B  899 -  900  "N7"  length=8.0  k=100     "23S 876- 879 w. 23S 899- 900; Doring et al."
crosslink   B  979 - 981, 983 - 984 "N7" WITH   B 2029 "N7"  length=8.0  k=100   "23S 979- 981, 983- 984 w. 23S 2029; Stiege et al."
crosslink   B 1018 - 1022      WITH   B 1140 - 1149        length=3.0  k=100     "23S 1018-1022 w. 23S 1140- 1149; Stiege et al."
crosslink   B 1210        "N7" WITH   B 1236         "N7"  length=8.0  k=100     "23S 1210 w. 23S 1236; Stiege et al."
crosslink   B 1325 - 1426      WITH   B 1574 - 1623        length=23.0 k=100     "23S 1325- 1426 w. 23S 1574- 1623; Turner & Noller"
crosslink   B 1433 - 1435      WITH   B 1556 - 1560        length=3.0  k=100     "23S 1433- 1435 w. 23S 1556- 1560; Stiege et al."
crosslink   B 1482        "N7" WITH   B 1501         "N7"  length=8.0  k=100     "23S 1482 w. 23S 1501; Stiege et al."
crosslink   B 1715        "N7" WITH   B 1743 - 1746  "N7"  length=8.0  k=100     "23S 1715 w. 23S 1743- 1746; Doring, refinement of Turner"
crosslink   B 1777 - 1792      WITH   B 2584 - 2588        length=3.0  k=100     "23S 1777-1792 w. 23S 2584-2588; Stiege et al."
crosslink   B 1782             WITH   B 2608 - 2609        length=3.0  k=100     "23S 1782 w. 23S 2608- 2609; Mitchell et al."
crosslink   B 1836 - 1839      WITH   B 1898 - 1903        length=3.0  k=100     "23S 1836- 1839 w. 23S 1898- 1903; Stiege et al."
crosslink   B 1911 - 1921 "N7" WITH   B 1964         "N7"  length=8.0  k=100     "23S 1911- 1921 w. 23S 1964; Doring et al."
crosslink   B 1933        "N7" WITH   B 1966         "N7"  length=8.0  k=100     "23S 1933 w. 23S 1966; Doring et al."
crosslink   B 1940             WITH   B 2554                length=3.0  k=100     "23S 1940 w. 23S 2554; Mitchell et al."
crosslink   B 1941 - 1942      WITH   B 1964 - 1965        length=3.0  k=100     "23S 1941- 1942  w. 23S 1964- 1965; Mitchell et al."
crosslink   B 1955             WITH   B 2552 - 2553        length=3.0  k=100     "23S 1955 w. 23S 2552-2553; Mitchell et al."
crosslink   B 2032        "N7" WITH   B 2054 - 2055  "N7"  length=8.0  k=100     "23S 2032 w. 23S 2054- 2055; Doring et al."
crosslink   B 2112        "N7" WITH   B 2169 - 2171  "N7"  length=8.0  k=100     "23S 2112 w. 23S 2169- 2171; Doring et al."
crosslink   B 2116 - 2117 "N7" WITH   B 2163 - 2167  "N7"  length=8.0  k=100     "23S 2116- 2117 w. 23S 2163- 2167; Doring et al."
crosslink   B 2128 - 2132 "N7" WITH   B 2156 - 2159  "N7"  length=8.0  k=100     "23S 2128- 2132 w. 23S 2156- 2159; Doring et al."
crosslink   B 2145 - 2146      WITH   B 2202                length=3.0  k=100     "23S 2145- 2146 w. 23S 2202; Mitchell et al."
crosslink   B 2257 - 2265      WITH   B 2425 - 2427        length=3.0  k=100     "23S 2257- 2265 w. 23S 2425- 2427; Stiege et al."
crosslink   B 2392 - 2393 "N7" WITH   B 2422 - 2423  "N1"  length=8.0  k=100     "23S 2392- 2393 w. 23S 2422- 2423; Doring et al."
crosslink   B 2518 - 2519      WITH   B 2544 - 2545        length=3.0  k=100     "23S 2518- 2519 w. 23S 2544- 2545; Mitchell et al."
crosslink   B 2737 - 2738 "N7" WITH   B 2763 - 2766  "N7"  length=8.0  k=100     "23S 2737- 2738 w. 23S 2763- 2766; Doring et al."
crosslink   B 2790 - 2791      WITH   B 2892 - 2895        length=3.0  k=100     "23S 2790- 2791 w. 23S 2892- 2895; Mitchell et al."
crosslink   B 2791        "N7" WITH   B 2890         "N7"  length=8.0  k=100     "23S 2791 w. 23S 2890; Doring et al."
crosslink   B 2832 - 2834      WITH   B 2878 - 2885        length=3.0  k=100     "23S 2832- 2834 w. 23S 2878- 2885; Stiege et al."
crosslink   B 2849 - 2852      WITH   B 2865 - 2867        length=3.0  k=100     "23S 2849- 2852 w. 23S 2865- 2867; Stiege et al."
```

Table C.2: 23S Cross-Links

```
crosslink   C 69   "N7" WITH   C 107  "N7"  length=8.0  k=100    "5S 69 w. 5S 107; Stiege et al."
```

Table C.3: 5S Cross-Links

```
// 16S = chainId A
// 23S = chainId B
crosslink   A 1408 - 1411   "N7" WITH   B 1912 - 1920  "N7"  length=8.0  k=100   "16S 1408- 1411 w. 23S 1912- 1920; Mittchel et al."
crosslink   A 1518 - 1520   "N7" WITH   B 1912 - 1920  "N7"  length=8.0  k=100   "16S 1518- 1520 w. 23S 1912- 1920; Mittchel et al."
```

Table C.4: Inter 16S/23S Cross-Links

```
// 23S = chainId B
//  5S = chainId C
crosslink   B  958          "C5" WITH   C   89              length=10.0 k=100   "23S 958 w. 5S 89; Sergiev et al."
crosslink   B  960          "C4" WITH   C   89              length=3.0  k=100   "23S 960 w. 5S 89; Dokudovskaya et al."
crosslink   B 1022          "C5" WITH   C   89              length=10.0 k=100   "23S 1022 w. 5S 89; Sergiev et al."
crosslink   B 1138          "C5" WITH   C   89              length=10.0 k=100   "23S 1138 w. 5S 89; Sergiev et al."
crosslink   B 2475          "C4" WITH   C   89              length=3.0  k=100   "23S 960 w. 5S 89; Dokudovskaya et al."
crosslink   B 2477          "C4" WITH   C   89              length=3.0  k=100   "23S 960 w. 5S 89; Dokudovskaya et al."
```

Table C.5: Inter 23S/5S Cross-Links

Summary

Corresponding to the aim of the thesis to improve the plausibility of a model of the ribosome, generated at the *Max-Planck-Institut für molekulare Genetik*, a computational approach of a more general character was designed and then applied to the specific problem. The complexity of the problem to be solved requires a multi-scale modelling approach with demands, that are almost contradictory, since local and global criteria can not meet the same degree of resolution.

The computational approach is based on the existing molecular dynamics technique which relies on the principles of classical physics and mechanics, instead of using quantum mechanics. It operates with Newtonian forces on atoms by replacing the quantum picture of interactions arising from overlapping electron clouds by a system of point masses, coupled by springs. This technique offers an efficient means for computing large scale structures. The inter-atomic relations are described by a set of forces, expressed as derivatives of an energy potential. Classical Newtonian mechanics allows the equations of motion of the atoms to be derived as well. By integrating these equations over time or by applying energy minimization algorithms on the potential function, stable structures can be calculated.

For solving the problem posed the standard molecular dynamics methods had to be extended, and new algorithms were designed. In brief, the steps leading to the main achievements in this computational part of the thesis are:

- The energy potential function, which is used in molecular dynamics to describe structural details of a particle, was extended to include terms that describe higher level biochemical constraints resulting from cross-linking techniques and cryo-electron microscopy (sections 1.2.4 and 1.2.5).

- By clustering defined groups of atoms into rigid objects, the number of parameters in this molecular dynamics approach was highly reduced, since only the positions and orientations of considerably fewer objects needed to be traced, instead of tracking many atomic positions. This reduction of parameters speeds up the computational process considerably (sections 1.5 and 1.6).

- The clustered molecular dynamics computational technique was integrated well into the interactive model building process by using the fact that the clustered rigid objects can correspond to the higher level structural blocks with which a biochemist operates in his reasoning (sections 1.5, 1.6 and 2.3.2).

- Software needed for the realization of these concepts was developed (section 1.7).

In order to adapt the computational methods described above to the ribosome as a specific particle the following steps were necessary:

- The extended energy potential mentioned above needed to be adapted to the structural properties of the ribosome.
  On a detailed level, it was specified which atoms are bound, what the desired bond lengths should be, which bond angles are important and which are the desired values of these angles; furthermore what the force constants are, and which Van-der-Waals effects matter.
  On a higher order level, the groups of atoms that can be clustered into rigid objects for the clustered molecular dynamics technique were specified.
  Additionally, schemes were developed which permit a particle of the complexity of the ribosome to be specified (chapter 5).

- The flow of information was organized in an iterative manner that integrates interactive, computational and feedback methods (chapter 6).

The general advantages of this approach are:

- The generated structural model is plausible, according to physicochemical demands (section 6.5).

- Higher level constraints given by cryo-electron microscopy (cf. section 4.3.4) and cross-linking techniques (cf. section 4.2.5) are incorporated into the model (section 6.5).

- Modifications of the overall structural model are achieved, including and maintaining precise sub-structures, investigated in detail by other techniques (section 6.5).

Following this access, a plausible structure of the ribosome results by refinements of the generated model mentioned above.

Bibliography

[BASS88] R. Brimacombe, J. Atmadja, W. Stiege, and D. Schüler. A detailed model of the three-dimensioanl structure of *E.coli* 16S ribosomal RNA in situ in the 30s subunit. *J. Mol. Biol*, 199:115–136, 1988.

[BBO+83] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, and M. Karplus. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry*, 4:187–217, 1983.

[BFN+98] N. Ban, B. Freeborn, P. Nissen, R.A. Grassucci, R. Sweet, J. Frank, P.B. Moore, and T.A. Steitz. A 9 Å resolution x-ray crystallographic map of the large ribosomal subunit. *Cell*, 93:1105–1115, 1998.

[Boh97] P. Bohner. Ein Multiagentenansatz zur Steuerung von redundanten Manipulatoren für medizinische Applikationen. Dissertation, Universität Karlsruhe (TH), Fakultät für Informatik, 1997.

[Bos93] S. Bosch. *Algebra.* Springer, Berlin, 1993.

[Bri84] R. Brimacombe. Conservation of structure in ribosomal RNA. *Trends in Biochem. Sciences*, 9:273–277, 1984.

[Bri98] R. Brimacombe. The ribosome – the universal protein-synthesizing machine. In *The Many Faces of RNA*, chapter 3, pages 41–53. Academic Press Ltd., 1998.

[CKEM88] M.S. Capel, M. Kjeldgaard, D.M. Engelman, and P.B. Moore. Positions of S2, S13, S16, S17, S19 and S21 in the 30S ribosomal subunit of *E.coli*. *J. Mol. Biol.*, 200:65–87, 1988.

[DM97] A. Dallas and P.B. Moore. The loop E-loop D region of *escherichia coli* 5S rRNA: the solution structure reveals an unusual loop that may be important for binding ribosomal proteins. *Structure*, 5(12):1639–1653, 1997.

[DW89] V. Drumm and W. Weil. Lineare Algebra und Analytische Geometrie. Skriptum, Universität Karlsruhe (TH), Mathematisches Institut II, 1989.

[Erc97] F. Ercolessi. A molecular dynamics primer. Technical report, International School for Advanced Studies (SISSA-ISAS), 1997.

[Erd96] T. Erdemir. Computergraphische Methoden für molekulare Strukturen in der Protein-Biosynthese. Dissertation, TU Berlin, FB Informatik, 1996.

[Fai85] R. Fairly. *Software Engineering Concepts.* McGraw-Hill, New York, 1985.

[FRBP96] D. Fourmy, M.I. Recht, S.C. Blanchard, and J.D. Puglisi. Structure of the a site of *E.coli* 16S ribosomal RNA complexed with an aminoglycoside antibiotic. *Science*, pages 1367–1371, 1996.

[FvDFH90] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics, Principles and Practice.* Addison-Wesley, Reading, Massachusetts, 1990.

[GGS93] R.R. Gutell, M.W. Gray, and M.N. Schnarre. A compilation of large subunit (23S- and 23S-like) ribosomal RNA structures. *Nucleic Acids Research*, 22:3055–3074, 1993.

[Gut94] R.R. Gutell. Collection of small subunit (16S- and 16S-like) ribosomal RNA structures. *Nucleic Acids Research*, 22:3502–3507, 1994.

[Ham43] Sir W.R. Hamilton. On a new species of imaginary quantities connected with a theory of quaternions. *Proceedings of the Royal Irish Academy*, 2:424–434, 1843.

[Ham44] Sir W.R. Hamilton. On quaternions. *Proceedings of the Royal Irish Academy*, 3 (1847):1–16, 1844.

[HH90] G. Hommel and H. Heiß. Roboterkinematik. Bericht 1990-15, TU Berlin, FB Informatik, 1990.

[HK86] B. Hardesty and G. Kramer. *Structure, function, and genetics of ribosomes.* Springer, New York, 1986.

[HM86] J.P. Hansen and I.R. McDonald. *Theory of simple liquids.* Academic Press, 2nd edition, 1986.

[Hog92] S.G. Hoggar. *Mathematics for Computer Graphics.* Cambridge University Press, Cambridge, 1992.

[HvH86] G. Harauz and M. van Heel. Exact filters for general geometry three dimensional reconstruction. *Optik*, 73:146–156, 1986.

[IEE83] *IEEE Standard Glossary of Software Engineering Terminology.* IEEE Standard 729-1983, 1983.

[JG95] P.Å. Jansson and R. Grahn. *Engineering Mechanics.* Prentice Hall, London, 1995.

[Joh85] L.N. Johnson. Protein crystallography. In *[NvD85]*, pages 347–415. Elsevier, 1985.

[KBW96] R. Koradi, M. Billeter, and K. Wüthrich. Molmol: a program for display and analysis of macromolecular structures. *J. Mol. Graphics*, 14:51–55, 1996.

[Lan93] S. Lang. *Algebra.* Addison-Wesley, Reading, Massachusetts, 1993.

[LB92] K.B. Lipkowitz and D.B. Boyd. *Reviews in Computational Chemistry III.* VCH, New York, 1992.

[LB93] K.B. Lipkowitz and D.B. Boyd. *Reviews in Computational Chemistry IV.* VCH, New York, 1993.

[Len98] Th. Lengauer. Bioinformatics – a different algorithmic experience. *invited talk at WAE*, 1998.

[MB97a] F. Mueller and R. Brimacombe. A new model for the three-dimensional folding of *Escherischia coli* 16S ribosomal RNA. I. fitting the RNA to a 3D electron microscopic map at 20 Å. *J. Mol. Biol.*, 271:524–544, 1997.

[MB97b] F. Mueller and R. Brimacombe. A new model for the three-dimensional folding of *Escherischia coli* 16S ribosomal RNA. II. the RNA-protein interaction data. *J. Mol. Biol.*, 271:545–565, 1997.

[MH87] J.A. McGammon and S.C. Harvey. *Dynamics of Proteins and Nucleic Acids.* Cambridge University Press, London, 1987.

[MH94] A. Malhotra and S.C. Harvey. A quantitative model of the *E.coli* 16S RNA in the 30S ribosomal subunit. *J. Mol. Biol.*, 240:308–340, 1994.

[MLS94] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robot Manipulation.* CRC Press, Boca Raton, 1994.

[MSB$^+$eda] F. Mueller, I. Sommer, P. Baranov, M. Rodnina, W. Wintermeyer, H. Stark, M. van Heel, and R. Brimacombe. The 3D arrangement of the RNA in the *Escherichia coli* 50S ribosomal subunit: I fitting the 23S and 5S rRNA to a cryoelectron microscopic map of the 70S ribosome at 13Å resolution. *J. Mol. Biol.*, 1999 (submitted).

[MSB$^+$edb] F. Mueller, I. Sommer, P. Baranov, M. Rodnina, W. Wintermeyer, H. Stark, M. van Heel, and R. Brimacombe. The 3D arrangement of the RNA in the *Escherichia coli* 50S ribosomal subunit: II distribution of functionally important sites. *J. Mol. Biol.*, 1999 (submitted).

[MSvH$^+$97] F. Mueller, H. Stark, M. van Heel, J. Rinke-Appel, and R. Brimacombe. A new model for the three-dimensional folding of *Escherischia coli* 16S ribosomal RNA. III. the topography of the functional centre. *J. Mol. Biol.*, 271:566–587, 1997.

[MTH94] A. Malhotra, R.K.Z. Tan, and S.C. Harvey. Modeling large RNAs and ribonuclein particles using molecular mechanics techniques. *Biophysical Journal*, 66:1777–1795, 1994.

[Mue93] F. Mueller. Entwicklung von Computergrafik-Methoden für die Darstellung der dreidimensionalen Struktur von ribosomalen Untereinheiten. Dissertation, TU Berlin, FB Informatik, 1993.

[NvD85] A. Neuberger and L. van Deenen. *Modern physical methods in biochemistry (New comprehensive biochemistry)*. Elsevier, New York, 1985.

[PN95] T. Powers and H.F. Noller. Hydroxil radical footprinting of ribosomal proteins on 16S rRNA. *RNA*, 1:194–209, 1995.

[PTVF97] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1997.

[Rap95] D.C. Rapaport. *The art of molecular dynamics simulation*. Cambridge University Press, Cambridge, UK, 1995.

[RJ85] J.K.M. Roberts and O. Jardetzky. Nuclear magnetic resonance spectroscopy in biochemistry. In *[NvD85]*, pages 1–67. Elsevier, 1985.

[RT89] H.P. Rehm and W. Trinks. Lineare Algebra und analytische Geometrie. Skriptum, Universität Karlsruhe (TH), Mathematisches Institut II, 1989.

[RW98] V. Ramakrishnan and S.W. White. Ribosomal protein structures: insights into the architecture, machinery and evolution of the ribosome. *TIBS*, 23:208–212, 1998.

[Sae84] W. Saenger. *Principles of Nucleic Acid Structure*. Springer, New York, 1984.

[SAZB86] W. Stiege, J. Atmadja, M. Zobawa, and R. Brimacombe. Investigation of the tertiary folding of *E.coli* ribosomal RNA by intra-RNA cross-linking in vivo. *J. Mol. Biol.*, 191:135–138, 1986.

[Sch92] Tamar Schlick. Optimization methods in computational chemistry. In *[LB92]*, pages 1–71. VCH, 1992.

[SJB95] K. Stade, N. Jünke, and R. Brimacombe. Mapping the path of the nascent peptide chain through the 23S RNA in the 50S ribosomal subunit. *Nucleic Acids Res.*, 23:2371–2380, 1995.

[SM95] A.A. Szewczak and P.B. Moore. The sarcin/ricin loop, a modular RNA. *J. Mol. Biol.*, 247:81–98, 1995.

[SOD+ss] M. Schatz, E.V. Orlova, P. Dube, H. Stark, F. Zemlin, and M. van Heel. Angular reconstitution in 3D electron microscopy: Practical and technical aspects. *Scanning Microscopy*, 1999 (in press).

[SRRA+97] H. Stark, M.V. Rodnina, J. Rinke-Appel, R. Brimacombe, W. Wintermeyer, and M. van Heel. Visualization of elongation factor Tu on the *Escherischia Coli* ribosome. *Nature*, 389:403–406, 1997.

[SSM86] G. Stöffler and M. Stöffler-Meilicke. Immuno electron microscopy on *E.coli* ribosomes. In *[HK86]*, pages 28–46. Springer, 1986.

[SWN88] S. Stern, B. Weiser, and H.F. Noller. Model for the three-dimensional folding of 16S ribosomal RNA. *J. Mol. Biol.*, 204:447–481, 1988.

[TNH+98] I. Tanaka, A. Nakagawa, H. Hosaka, S. Wakatsuki, F. Mueller, and R. Brimacombe. Matching the crystallographic structure of ribosomal protein S7 to a three-dimensional model of the 16S ribosomal RNA. *RNA*, 4:542–550, 1998.

[vH84] M. van Heel. Multivariate statistical classification of noisy images (randomly oriented biological macromolecules). *Ultramicroscopy*, 13:165–184, 1984.

[vH87] M. van Heel. Angular reconstitution: A posteriori assignment of projection directions for 3D reconstruction. *Ultramicroscopy*, 21:111–124, 1987.

[vHOH+ss] M. van Heel, E. Orlova, G. Harauz, H. Stark, P. Dube, F. Zemlin, and M. Schatz. Angular reconstitution in 3D electron microscopy: Historical and theoretical aspects. *Scanning Microscopy*, 1999 (in press).

[vLEH99] M.S. van Loock, T.R. Easterwood, and S.C. Harvey. Major groove binding of the tRNA/mRNA complex of the 16S ribosomal RNA decoding site from *Escherichia coli*. *J. Mol. Biol.*, 285:2067–2078, 1999.

[Wal90] W. Walter. *Gewöhnliche Differentialgleichungen*. Springer, Berlin, 1990.

[WGM+99] B.T. Wimberly, R. Guymon, J.P. McCutcheon, J.P. White, and V. Ramakrishnan. A detailed view of a ribosomal active site: the structure of the GTPase center at 2.6Å resolution. *Cell*, 97:491–502, 1999.

[Wit83] H.G. Wittmann. Architecture of prokaryotic ribosomes. *Ann. Rev. Biochem.*, 52:35–65, 1983.

[Wit85] H.G. Wittmann. Structure of ribosomes. In *[HK86]*, pages 1–27. Springer, 1985.

[YF98] A. Yonath and F. Franceschi. Functional universality and evolutionary diversity: insights from the structure of the ribosome. *Structure*, 6:679–684, 1998.

Index